Miten hyödyntää ternäärisiä tietokoneita?

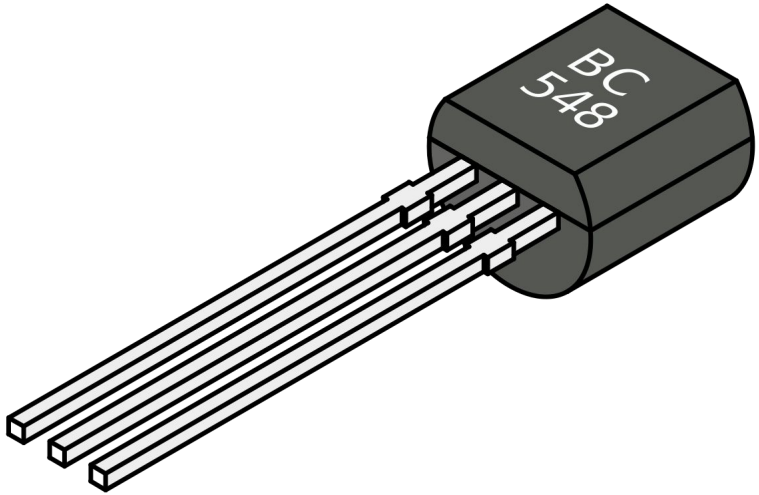
Rasmus Suonio, Lassi Niemelä 2020-2021

Sisällysluettelo

Johdanto

Klassisen tietokoneen toiminnasta

Klassinen tietokone käyttää laskemiseen binäärijärjestelmää. Tämä helpottaa transistorien rakentamista, sillä numeroa 0 voidaan kuvata katkaistuna virtana ja numeroa 1 kulkevana virtana. Tietokone ratkaisee ongelmia AND-, OR- ja NOT-porteilla. Portteja yhdistelemällä voidaan saada NAND, NOR, XOR ja XNOR portteja. (Lähde: https://en.wikipedia.org/wiki/Computer)



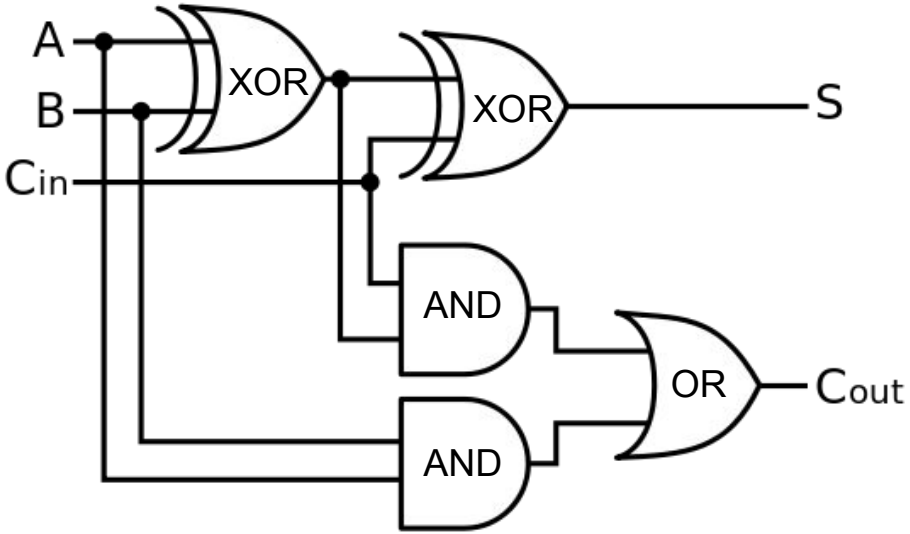
Syöte		Tuloste					
A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Miten klassinen tietokone suorittaa yhteenlaskun?

Klassinen tietokone laskee yhteenlaskuja kuvan mukaisella tavalla. Yleisesti on vain 2 syötettä (A ja B)

(Lähde: https://en.wikipedia.org/wiki/Adder_(electronics))

Periaatteessa ainoat asiat mitä tietokone tekee on yhteenlasku, vähennyslasku ja arvojen vertaaminen (XNOR)



Ternäärinen tietokone

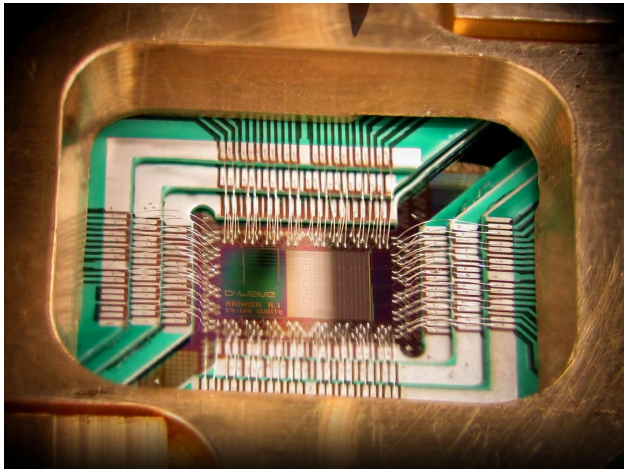
Ternäärisen tietokoneen historia: Setun

Ensimmäinen ternäärinen tietokone, Setun rakennettiin Neuvostoliitossa vuonna 1958. Se ratkaisi tieteellisiä ja teknisiä pulmia, ja oli erittäin tehokas ratkaisemaan matemaattisia pulmia ja mallintamaan muun muassa fysiikkaa ja kemiaa matemaattisesti. Setunissa oli 81 sanan kattava muisti, joka koostui 18 tritistä. Setun oli sekä halvempi, että tehokkaampi, kuin normaali binääri tietokone. Setunia oli tarkoitus massatuottaa, mutta Neuvostoliiton elektroniikkaministeriö ei hyväksynyt hanketta.

(Lähde: https://en.wikipedia.org/wiki/Setun)

Ternäärinen tietokone-Kvanttitietokone synergia

Kvanttitietokoneessa on kolme eri tilaa: 0, 1 tai jotain siltä väliltä. Ternäärisessä tietokoneessa on 3 arvoa. Samankaltaisuuksien takia ternäärinen tietokone voisi todennäköisesti toimia tehokkaammin kvanttitietokoneen kanssa kuin binääriset tietokoneet. Kvanttitietokone kuitenkaan tuskin koskaan tulee olemaan yksityishenkilöillä käytössä, mutta jos alettaisiin käyttää kvanttiservereitä (oma ideamme, jossa kvanttitietokone toimii palvelimena), ternäärinen tietokone voisi korvata klassiset tietokoneet, sillä se olisi hyvin todennäköisesti tehokkaampi kvanttiserverin kanssa kommunikoimisessa.



Miksi ternäärinen tietokone?

Kolmoisjärjestelmä on tehokkain tapa merkitä numeroita, sillä viereinen kaava on mahdollisimman pieni, jos luku n on e eli noin 2,718. Kuitenkin tietokoneilla järjestelmän on pakko olla kokonaisluku, joten paras järjestelmä on 3. Klassisen tietokoneen 64 bittiä voidaan esittää 41 ‘tritillä’. Meidän mielestämme optimaalinen tryten (ternääri tavun) koko on 6 trittiä, eli “hextritti”. Seuraavissa kalvoissa tulemme selittämään miksi.

$$\lim_{x \to \infty} log_n(x) \times n$$

Jatkossa viittaamme tähän kantalukujärjestelmän kykyyn merkitä numeroita tehokkaasti tehokkuutena. Jos siis sanomme, että ternäärinen tietokone on tehokkaampi, tarkoitamme, että kolmoisjärjestelmä on tehokkaampi tapa merkitä numeroita kuin binääri.

Selitys ternäärisen tietokoneen matematiikasta

$$\lim_{x \to \infty} log_n(x) \times n$$

Kaavan lähde:
https://www.jstor.org/stable/27857554?seq=1

Huomio: Kun tietokonetta fyysisesti tehdään, width, eli $\log_n(x)$, on kokonaisluku. Se takia pyöristämme tässä.

Viereinen kaava tarkoittaa luvun merkitsemiseen tarvittavien numeroiden määrää. Esimerkiksi kymmenjärjestelmässä luvun 221 esittämiseen tarvitaan 10 mahdollista numeroa, eli n (esityksessä puhutaan myös r:stä) = 10. Numeroita on 3, eli vastaukseksi tulee 3*10, joka on 30.

Vastaavasti jos sama tehtäisiin kolmoisjärjestelmässä, kantaluku (n tai r) olisi 3, ja width, “leveys” on tällöin logaritmi järjestelmässä kolme kyseisestä luvusta 221, eli $\log_3(221)$, joka on noin 4,9 eli pyöristettynä 5. Tällöin r*w, kantalukujärjestelmä kerrottuna “leveydellä” eli tarvittavalla potenssilla on 3*5 = 15. Jo näin pienellä luvulla kantalukujärjestelmän vaikutus tehokkuuteen on huomattavaa.

Vastaavasti binäärissä luvun muodostamiseen tarvitaan 8 numeropaikkaa, eli width = 8 ja 2 mahdollista numeroa, eli rw = 2*8=16

Tästä lähtien viittaamme kyseiseen kaavan rw:nä..
Esimerkiksi seuraavassa kalvossa sanotaan “Todistetaan, että rw on…”

Matemaattinen todiste 1: Todistetaan, että rw on pienempi kolmoisjärjestelmässä kuin binäärissä isoilla luvuilla.

1. $3 * (log_3(x) + 1) < 2 * (log_2(x) + 1)$
2. $3^{3*(log_3(x)+1)} < 3^{2*(log_2(x)+1)}$
3. $27x^3 < 2^{2*(log_2(x)+1)} * 1.5^{2*(log_2(x)+1)}$
4. $27x^3 < 4x^2 * 1.5^{2*(log_2(x)+1)}$
5. $27x < 4 * 1.5^{2*log_2(x)} * 1.5^2$
6. $3x < 1.5^{2*log_2(x)}$
7. $3x < 0.75^{2*log_2(x)} * 2^{2*log_2(x)}$
8. $3x < (0.75^{log_2(x)})^2 * x^2$
9. $3 < 2^{log_2(x)* log_2(0.75)})^2 * x$
10. $3 < x^{2*log_2(0.75)} * x$
11. $3 < x^{1+2*log_2(0.75)}$
12. $3 \lesssim x^{0.18}$

Selitys:

1. Oletetaan, että rw luvulla 3 on pienempi kuin rw luvulla 2.
2. 3 korotetaan rw:n potessiin kummallakin puolella
3. Ratkaistaan vasen puoli ja oikea puoli hajotetaan kahten osaan
4. Järjestellään lukuja ja jaetaan molemmat puolet x^2:lla.
5. Hajotetaan oikean puolen toinen termi ja jaetaan 9:llä
6. Tilanne on nytten tämä.
7. Hajotetaan jälleen kahteen osaan.
8. Jaetaan x:llä
9. Tehdään hieno taikatemppu
10. Päädytään tähän tulukseen,
11. ja esitetän tulos yhtenä terminä
12. Annetaan likiarvo.

Liitetiedostossa “Kuvaaja 2” näkyy kaavan kuvaajat.

Matemaattinen todiste 2: Todistetaan, että rw:n derivaatta on pienempi kolmoisjärjestelmässä

1. $\frac{d}{dx}(2 * (log_2(x) + 1) \quad > \quad \frac{d}{dx}(3 * (log_3(x) + 1)$

Koska $\frac{d}{dx}\ log_n(x) = \frac{1}{x*ln(n)}$ ja $\frac{d}{dx}C = 0$ sekä $\frac{d}{dx}f(x) * g(x) = f'(x) * g(x) + f(x) * g'(x)$, voimme laskea noiden sääntöjen avulla näin:

2. $\frac{2}{x*ln(2)} > \frac{3}{x*ln(3)}$

Jos kerromme molemmat puolet $x * ln(2) * ln(3)$:lla, tulokseksi tulee

3. $2 * ln(3) > 3 * ln(2)$, jos x > 0.

Jos e, eli noin 2.718 korotetaan noihin potensseihin, siitä tulee

4. $e^{2*ln(3)} > e^{3*ln(2)}$
5. $3^2 > 2^3$
6. $9 > 8$
7. Tosi.

Liitetiedostossa “Kuvaaja 2” näkyy kuvaajat kyseisille derivaatan tuloksille.

Jos haluat tarkemmat selitykset kaikille kolmelle matemaattiselle todisteelle, avaa liitetiedosto “**Matemaattinen todiste**”.

Tässä otamme derivaatan, eli käytännössä kulmakertoimen aiemmasta kaavasta $log_n(x)*n$, arvoilla 2 ja 3. Lähdemme siitä oletuksesta, että kantaluvulla 3 tietokoneen kaava kasvaa hitaammin. Otamme derivaatat, ja päädymme tulokseen, että se on aina tosi jos $r \neq 0$

Matemaattinen todiste 3: Lasketaan derivaattojen suhde, eli kuinka paljon nopeammin binäärissä rw kasvaa kuin kolmoisjärjestelmässä

$$rw'(2) = \frac{2}{x*ln(2)}$$

$$rw'(3) = \frac{3}{x*ln(3)}$$

Lasketaan $\frac{rw'(2)}{rw'(3)}$.

$$\frac{rw'(2)}{rw'(3)} = \frac{\frac{2}{x*ln(2)}}{\frac{3}{x*ln(3)}} = \frac{2*ln(3)}{3*ln(2)}$$

Jos haluat tarkemmat selitykset kaikille kolmelle matemaattiselle todisteelle, avaa liitetiedosto “**Matemaattinen todiste**”.

Huom. Kun tämän tarkan arvon laskee, huomaa, että binäärisen tietokoneen “tehottomuus”, eli rw kasvaa aina 5,66% nopeammin kuin ternäärisen tietokoneen.

Miten ternäärinen tietokone käytännössä toimisi?

Laskennassa voitaisiin käyttää erisuuruisia jännitteitä tai virtoja kuvaamaan kolmea eri tilaa: 0 - ei virtaa tai jännitettä, 1 - pieni virta tai jännite ja 2 - suuri virta tai jännite. Ongelmia tässä laskutavassa ovat mikropiirien koko ja valmistus. Ternäärinen tietokone voisi käyttää yhteenlaskuun kahta porttia (C ja R. Löytyy seuraavasta kalvosta)

Huom. Tutkimukseemme ei kuulu tutkia, miten ternäärinen tietokone käytännössä toimisi, mutta tässä kuitenkin on muutamia ideoita. Tiedämme, että ternäärinen tietokone on huomattavasti hankalampi rakentaa, emme ota huomioon fysiikkaa, sillä tämä on matemaattinen tutkimus

C

Ternäärisen tietokoneen esimerkkiportit, joiden avulla voisi saada ternäärisen tietokoneen laskemaan yhteen- ja vähennyslaskuja.

R

SYÖTE	0	1	2
0	0	0	0
1	0	0	1
2	0	1	1

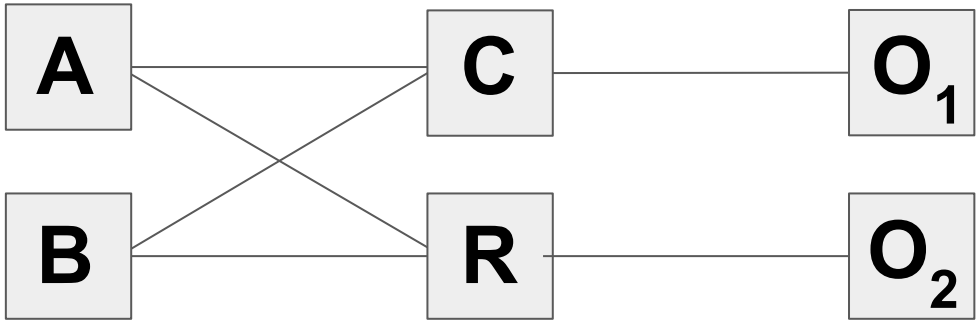
SYÖTE	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Esimerkkipiiri ternäärisen tietokoneen yhteenlaskusta

Funktiot:

```
def power(x, y):
    if x != 0 and y != 0:
        return True
def resistor(z, w):
    return z/w
def comparator(x, y):
    if x >= y:
        return True
    else:
        return False
```

Tällä tavalla C ja R -porteilla voi laskea yhteenlaskun. Ongelmaksi kuitenkin tulee miten rakentaa kyseiset portit fyysisesti. Ohessa näkyy Python-koodi, joka rakentaa C ja R-portit käyttäen kolmea elektronista porttia: Binääristä and-gatea, vastusta ja binääristä komparaattoria.



O_1 ja O_2 ovat outputit, eli ulostulot.

```
def R(x, y):
    if not power(x, x):
        return y
    elif not power(y, y):
        return x
    elif power(x, y):
        if comparator(resistor(x, 2), y) or comparator(resistor(y, 2), x):
            return 0
        elif comparator(resistor(x, 2), 1) and comparator(resistor(y, 2), 1):
            return 1
        else:
            return 2
def C(x,y):
    if power(x, y) and (comparator(resistor(x, 2), 1) or comparator(resistor(y, 2), 1)):
        return 1
    else:
        return 0
while True:
    x = int(input("Syöte 1 (x): "))
    y = int(input("Syöte 2 (y): "))

    print(str(C(x,y)), str(R(x, y)))
```

```python
from timeit import default_timer as timer
import random
x = random.randint(0,100)
y = random.randint(0,100)
def comparison(x, y):
    if x == y:
        pass
    elif x > y:
        pass
    elif x < y:
        pass
start = timer()
if x == y:
    pass
elif x > y:
    pass
elif x < y:
    pass
end = timer()
print((end - start)*1000000, " mikrosekuntia")
start = timer()
comparison(x, y)
end = timer()
print((end - start)*1000000, " mikrosekuntia")
```

Päätimme myös testata ternäärisen tietokoneen nopeutta käytännössä. Se on tietenkin mahdotonta binäärisellä tietokoneella, mutta teimme parhaamme. Vieressä näkyy koodi, jota käytimme. Aluksi luomme pseudorandomit vakiot x ja y. Sitten määrittelemme funktion, joka tarkastaa luvut. Tämä siis toimii epätäydellisenä ternääriporttina. Sen jälkeen koodissa teemme saman binäärissä, ja mittaamme, kuinka kauan kummankin suorittamisessa kestää. Kokeilimme moneen kertaan, ja jokaisella kerralla ternäärifunktio oli ainakin kaksi kertaa nopeampi kuin binäärifunktio. Tämä ei ole tietenkään täydellinen koe, eikä sitä voi pitää yksinään todisteena ternäärisen tietokoneen nopeudesta. Kuitenkin se varmistaa jo aikaisemmin tekemämme havainnot.

Pseudorandom = tietokoneella luotu satunnaisluku
Huom: Kaikki koodit löytyvät liitetiedostoista .py-tiedostomuodossa.

Tulokset:

```
1.6000000000043757  mikrosekuntia
0.6999999999993123  mikrosekuntia
```

Kuvat

Kuva transistorista (kalvo 2) Löytyy osoitteesta:
https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/TO-92%2C_BC548_(front%2C_shaded).svg/1200px-TO-92%2C_BC548_(front%2C_shaded).svg.png
Kuva full adder (kalvo 4) Löytyy osoitteesta:
https://upload.wikimedia.org/wikipedia/commons/thumb/a/a9/Full-adder.svg/550px-Full-adder.svg.png
Kuva ternäärinen tietokone Setun (kalvo 9 ja 11) Löytyy osoitteesta:
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSzfnIXueUczYvarxCuz1WssqhtA8U0O5-gwkdPSVKQVrob1gIdzyiIOqlTtbAw8TZDchg&usqp=CAU
Kvanttitietokone (kalvo 11) Löytyy osoitteesta:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fcommons.wikimedia.org%2Fwiki%2FFile%3ADWave_128chip.jpg&psig=AOvVaw2LNd80J5_fLDJLX7lpI_3x&ust=1608042527772000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCJCX4LbXze0CFQAAAAAdAAAAABAD

Tietolähteet

Jstor: Brian Hayes: “Computing Science: Third Base” 12/2001 Löytyy osoitteesta: https://www.jstor.org/stable/27857554?seq=1
Wikipedia: Binäärijärjestelmä. Löytyy osoitteesta: https://fi.wikipedia.org/wiki/Binäärijärjestelmä
Wikipedia: Logic gate. Löytyy osoitteesta: https://en.wikipedia.org/wiki/Logic_gate
Wikipedia: Computer. Löytyy osoitteesta: https://en.wikipedia.org/wiki/Computer
Wikipedia: Adder. Löytyy osoitteesta: https://en.wikipedia.org/wiki/Adder_(electronics)#Full_adder
www.computer-museum.ru: МЦВМ Сетунь Löytyy osoitteesta: https://www.computer-museum.ru/histussr/12-6.htm
Springer.com: Ternary Computers: The Setun and the Setun 70 Löytyy osoitteesta: https://link.springer.com/content/pdf/10.1007%2F978-3-642-22816-2_10.pdf
Wikipedia: Setun Löytyy osoitteesta: https://en.wikipedia.org/wiki/Setun