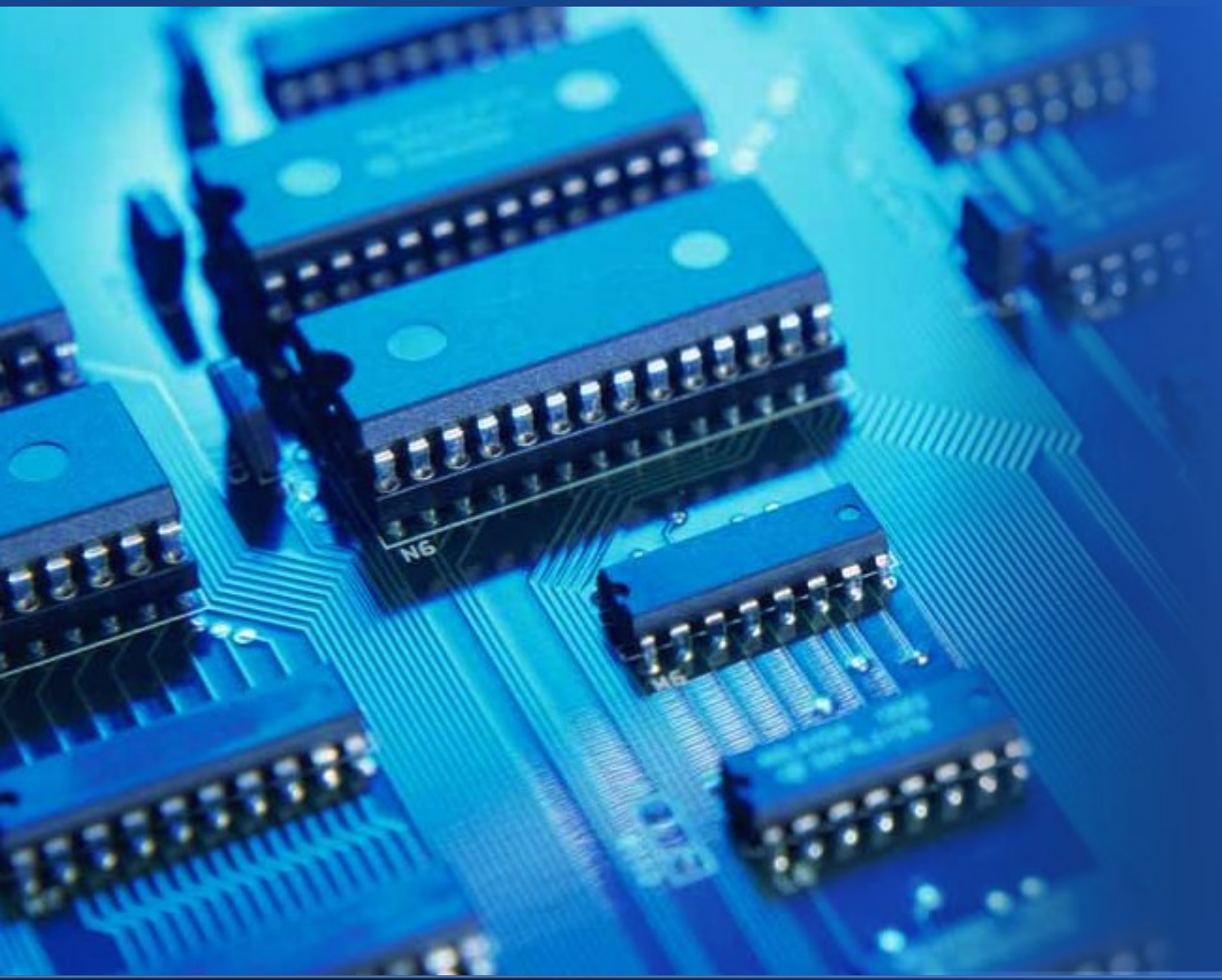
Digitaalitekniikan alkeiskurssi



Sekvenssi-logiikka

Helsinki Hacklab

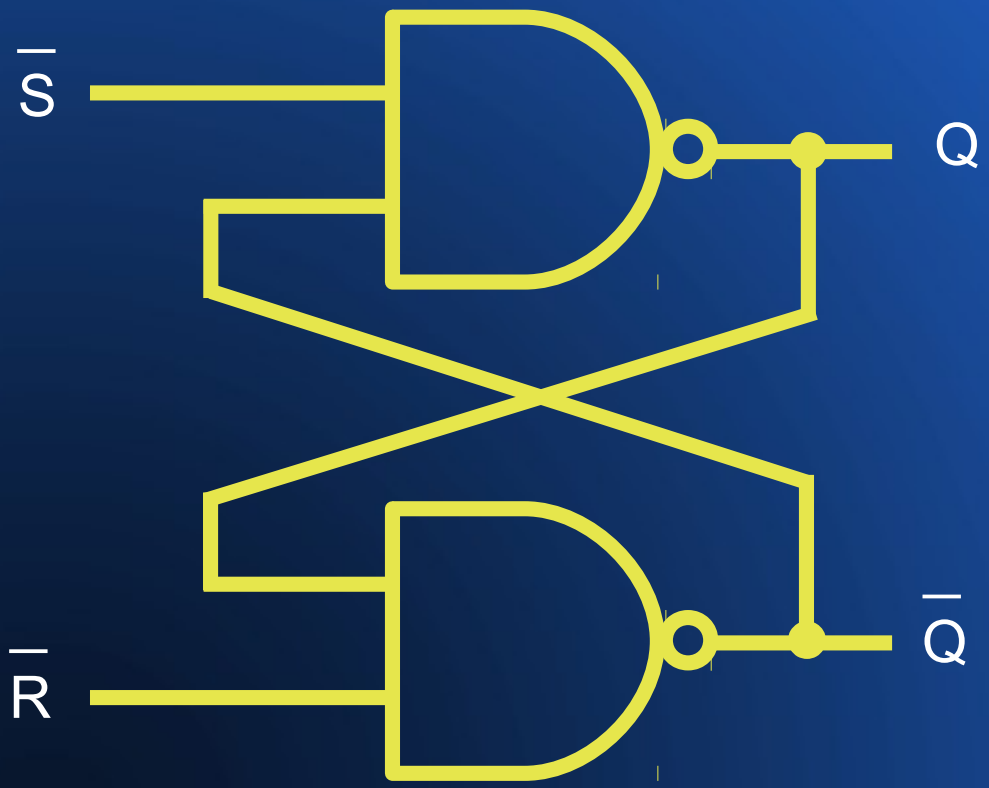
Logiikkojen jaottelu

- Kombinaatiologiikka
  - Lähtöjen tilat riippuvat ainoastaan tämän hetkisistä tulojen tiloista
- Sekvenssilogiikka
  - Lähtöjen tilat riippuvat sekä tämän hetkisistä että aikaisemmista tulojen tiloista

Sekvenssilogiikat

- Kiikut/latchit
  - RS, JK, D, monostabiili
  - addressable
- Laskurit
  - ripple carry, synchronous
  - ÷10, ÷16, …
  - up/down, presettable
- Siirtorekisterit
  - serial in, parallel out
  - parallel in, serial out

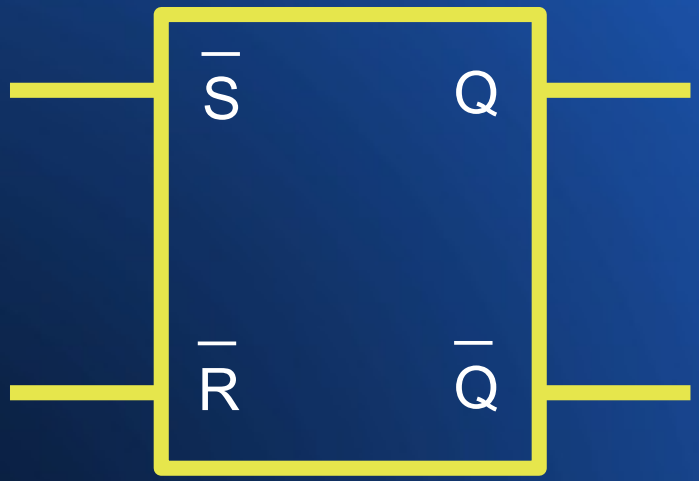
RS-kiikku NANDeilla



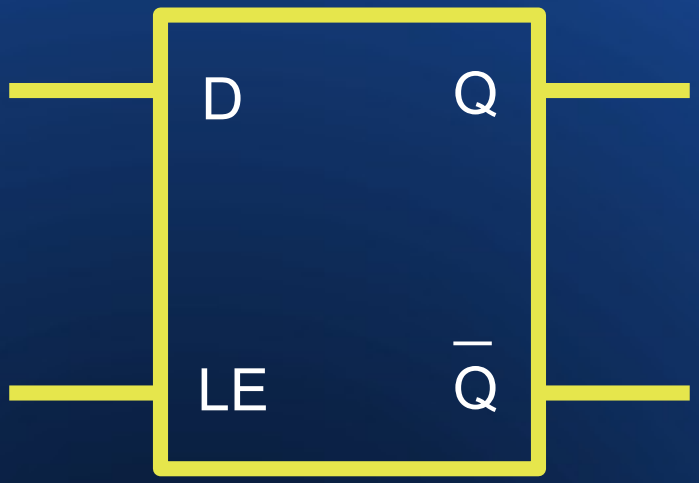
NAND-portin totuustaulu:

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

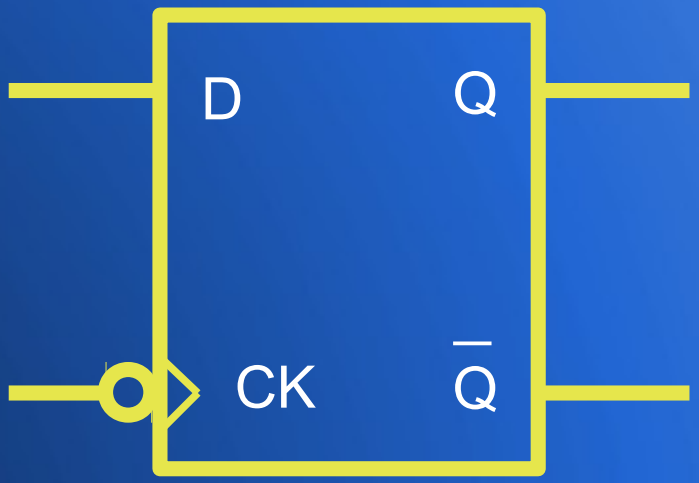
RS-kiikku symbolina



D-kiikku

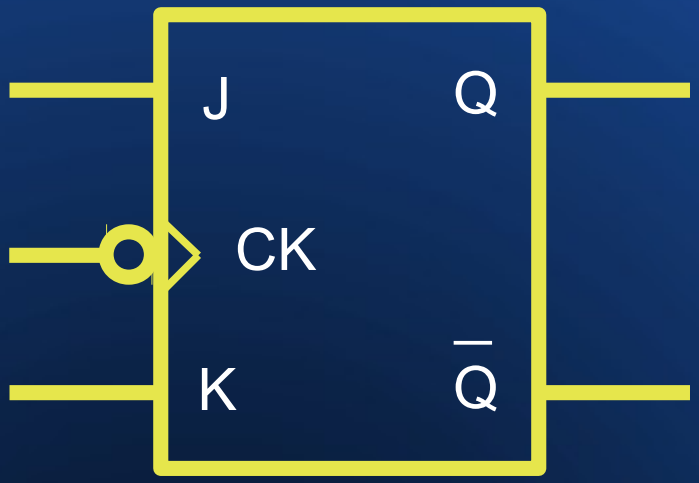


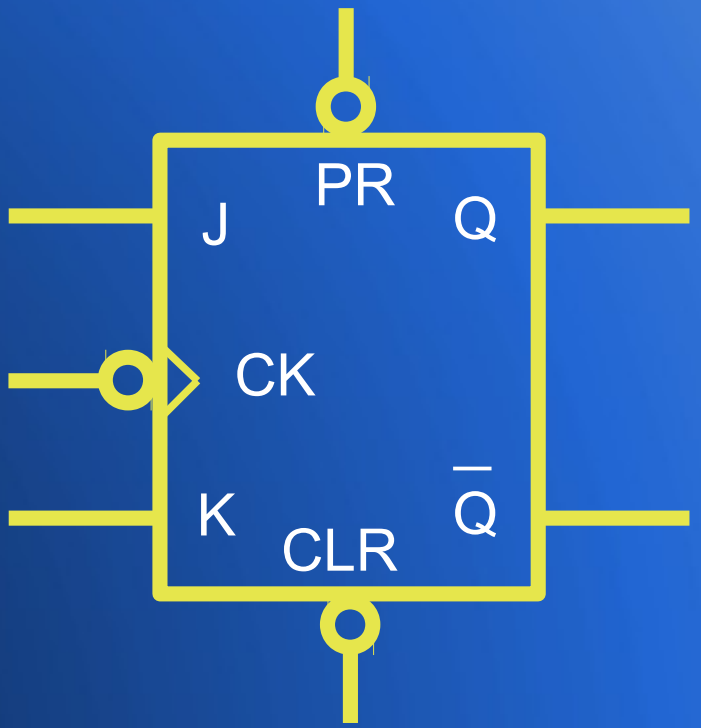
Läpinäkyvä latch



Laskevalla reunalla
liipaistava D-flip-flop

JK-kiikku



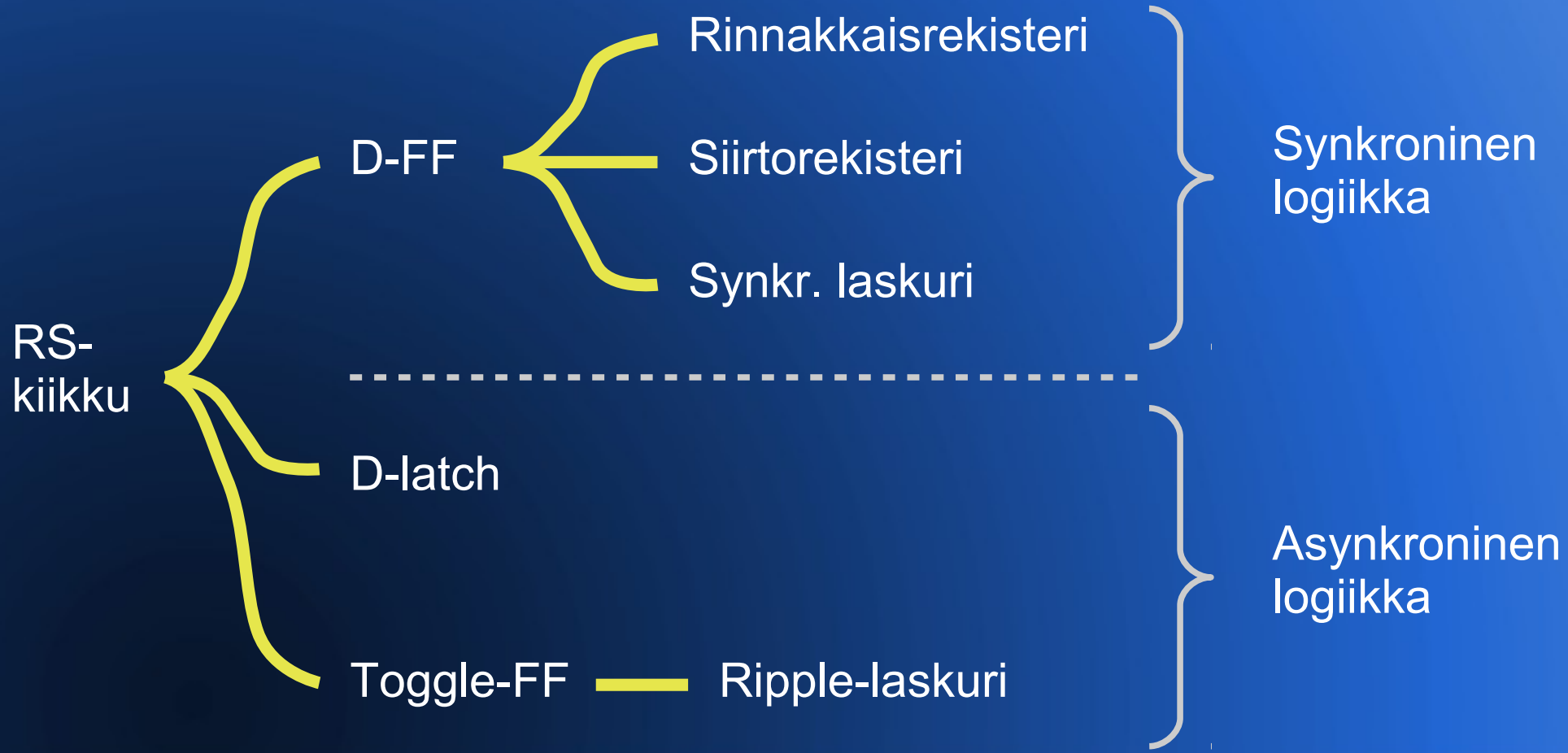


JK with preset and clear

Esimerkkejä

- 4-bittinen binäärilaskuri, ripple carry (74xx93)
  - play-hookey: basic 4-bit counter
- 4-bittinen binäärilaskuri, synkroninen (74xx163)
  - play-hookey: synchronous binary counter
- Siirtorekisteri: sarja sisään, rinnan ulos (74xx595)
  - play-hookey: shift register (S to P)
- Siirtorekisteri: rinnan sisään, sarja ulos (74xx165/166)
  - play-hookey: shift register (P to S)

Sekvenssilogiikan “sukupuu”



Digitaalitekniikan alkeiskurssi

Sekvenssi-logiikka

Helsinki Hacklab

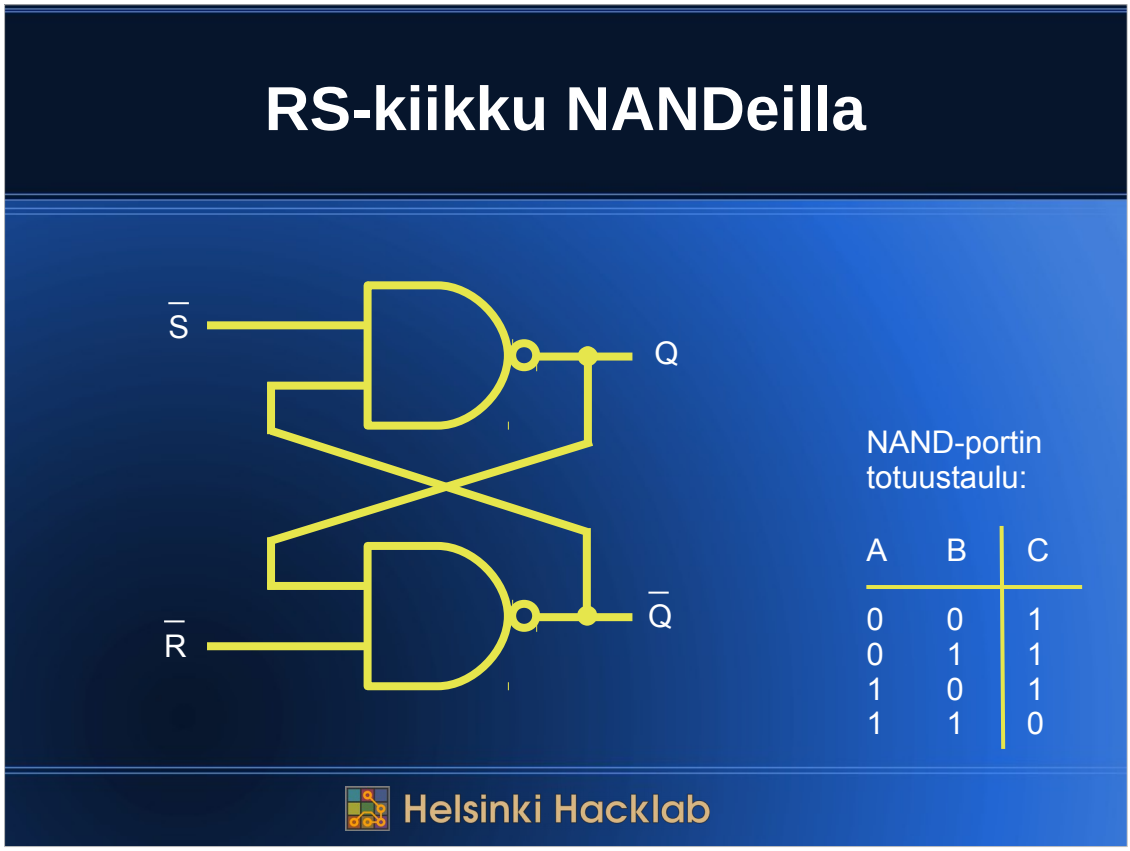
Kombinaatio- ja sekvenssilogiikat

Logiikkojen jaottelu

- Kombinaatiologiikka
  - Lähtöjen tilat riippuvat ainoastaan tämän hetkisistä tulojen tiloista
- Sekvenssilogiikka
  - Lähtöjen tilat riippuvat sekä tämän hetkisistä että aikaisemmista tulojen tiloista

Sekvenssilogiikat

- Kiikut/latchit
  - RS, JK, D, monostabiili
  - addressable
- Laskurit
  - ripple carry, synchronous
  - ÷10, ÷16, …
  - up/down, presettable
- Siirtorekisterit
  - serial in, parallel out
  - parallel in, serial out

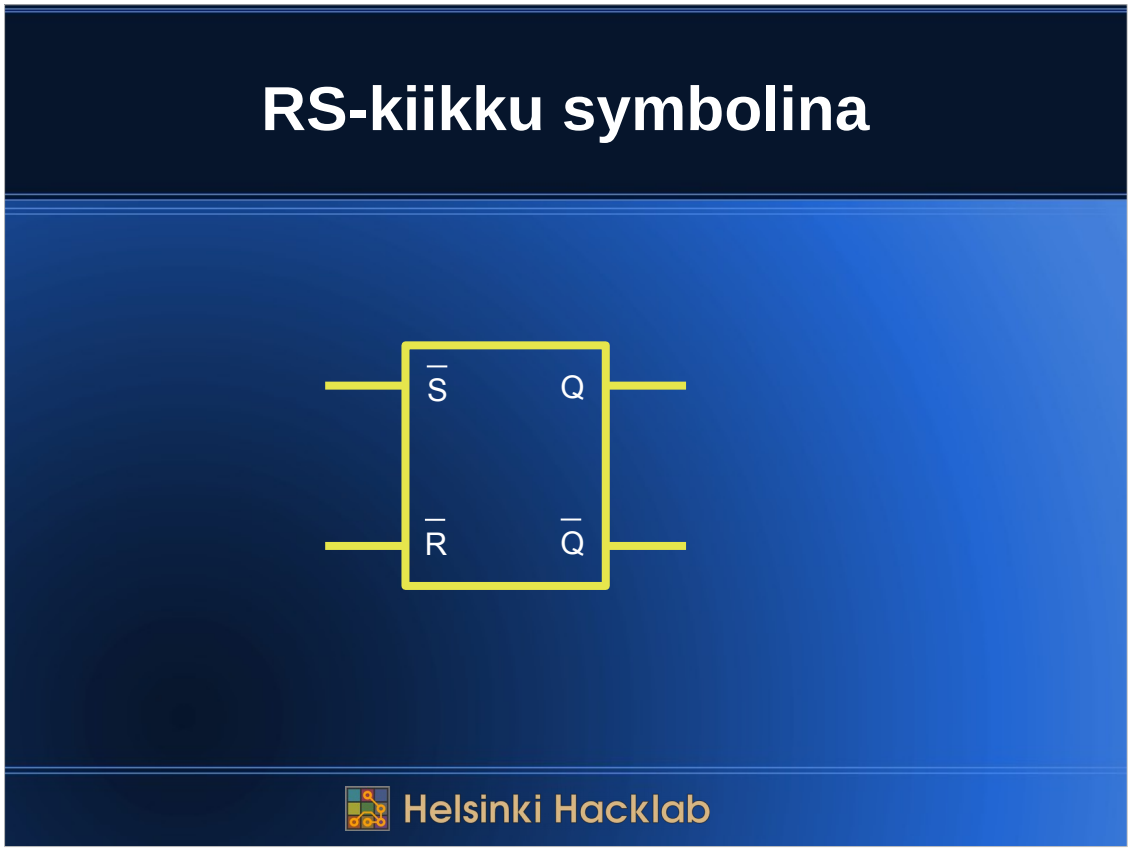


Perus-RS-kiikku toteutettuna kahdella NAND-piirillä. Kytkennällä on kaksi tilaa ja se pysyy niistä kummassa tahansa.

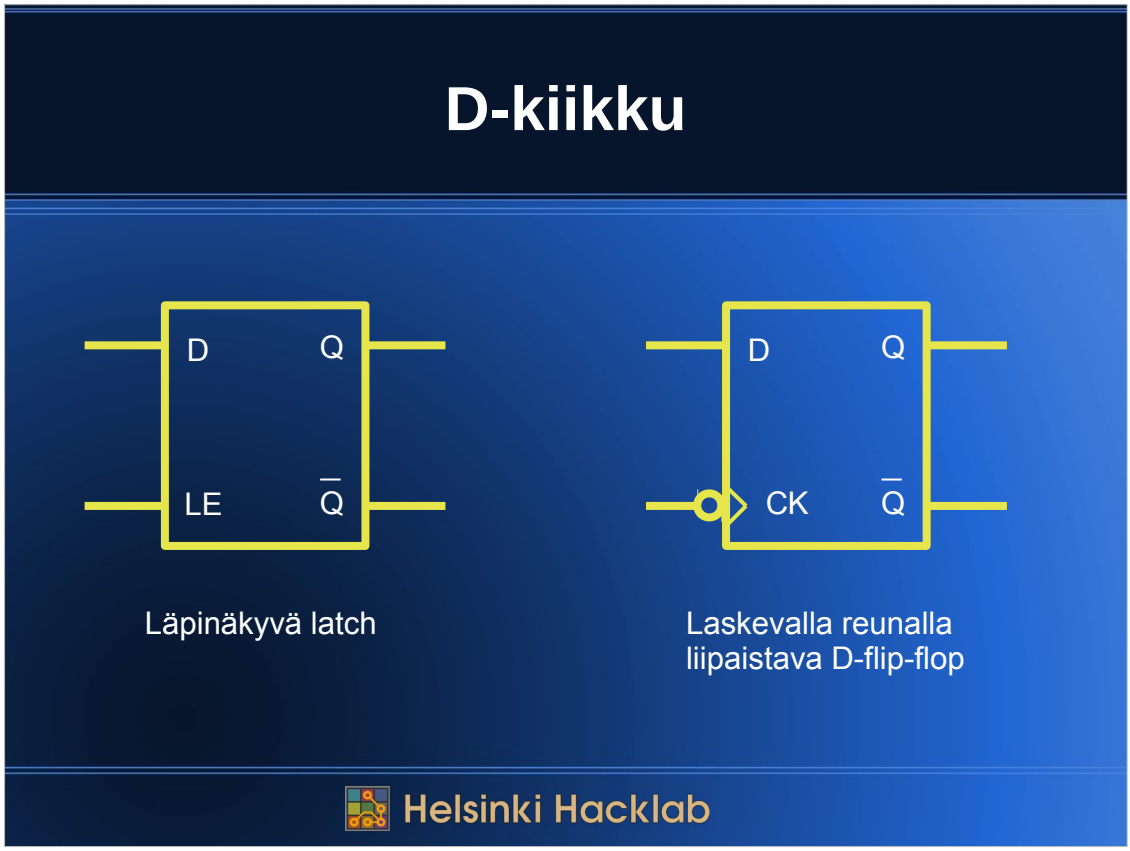
Toisessa tilassa lähtö Q=L ja toisessa Q=H. Lähtö $\overline{Q}$ on aina päinvastaisessa tilassa kuin Q.

Normaalisti $\overline{S}$ (set) ja $\overline{R}$ (reset) ovat H-tilassa. Kun $\overline{S}$ käy L-tilassa Q asettuu ja jää asettuneeksi. Kun $\overline{R}$ käy L-tilassa, Q nollaantuu ja jää nollaantuneeksi.

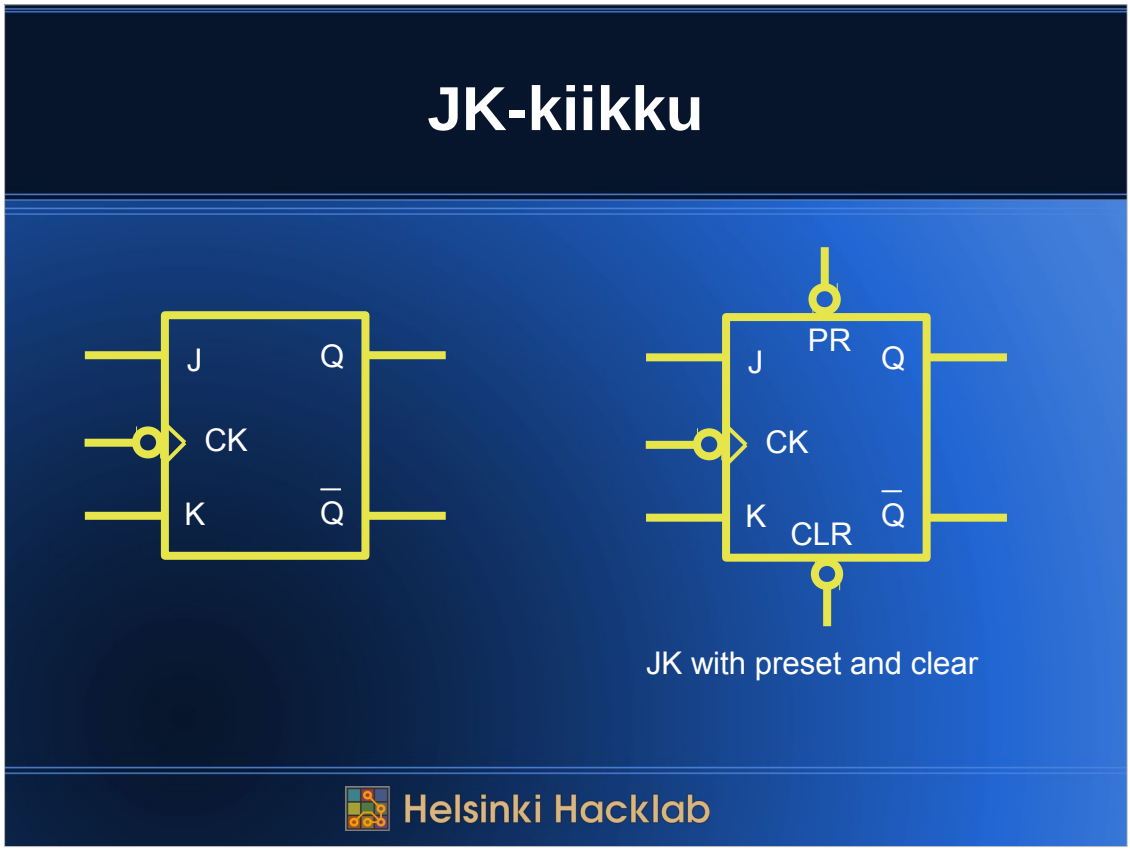
Tila, jossa $\overline{S}$ ja $\overline{R}$ ovat yhtäaikaa L-tilassa on kielletty. Tämän tilan aika molemmat lähdöt ovat H-tilassa. Kielletyn tilan poistuttua lähdön Q-tila on satunnainen.



Normaalisti kiikut, myös RS-kiikku, piirretään symbolina, joka kuvaa sen toiminnan, ei sisäistä porttitason rakennetta.



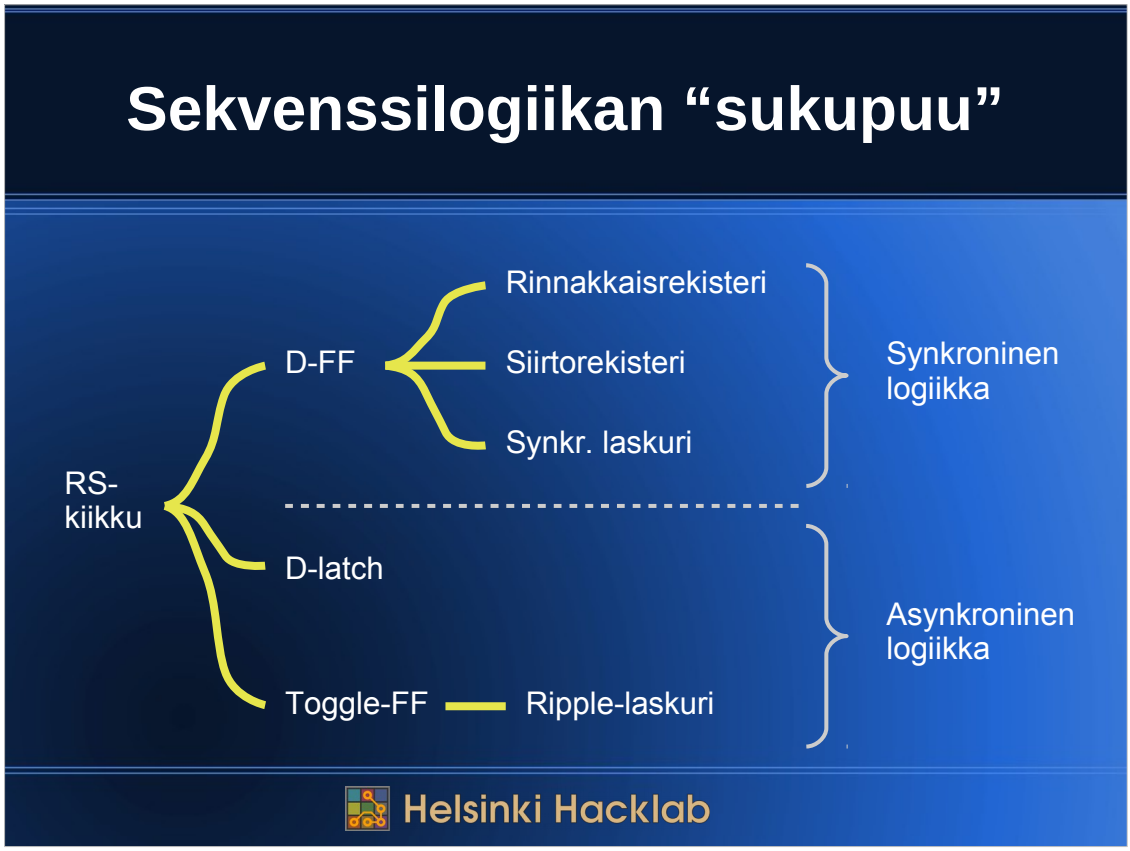
Erilaisia D-kiikkuja. Tarkemmin play-hookey.com.



Erilaisia JK-kiikkuja. Tarkemmin play-hookey.com.

Esimerkkejä

- 4-bittinen binäärilaskuri, ripple carry (74xx93)
  - play-hookey: basic 4-bit counter
- 4-bittinen binäärilaskuri, synkroninen (74xx163)
  - play-hookey: synchronous binary counter
- Siirtorekisteri: sarja sisään, rinnan ulos (74xx595)
  - play-hookey: shift register (S to P)
- Siirtorekisteri: rinnan sisään, sarja ulos (74xx165/166)
  - play-hookey: shift register (P to S)



Synkronisessa logiikassa kaikki D-flip-flopit saavat saman kellopulssin eli liipaistuvat samalla hetkellä. Koko hommaa tahdittaa yhteinen kello-oskillaattori.

Asynkronisessa logiikassa flip-flopin kello voi tulla jonkun toisen piirin lähdöstä, eikä kaikille yhteistä kellogeneraattoria ole välttämättä ollenkaan.

Jatkossa keskitytään lähes pelkästään synkroniseen logiikkaan, koska CPLD- ja FPGA-piirit on tarkoitettu synkronisen logiikan toteuttamiseen. CPLD:llä voi tietyin rajoituksin toteuttaa asynkronistakin logiikkaa, FPGA:lla se ei yleensä ole järkevää.