

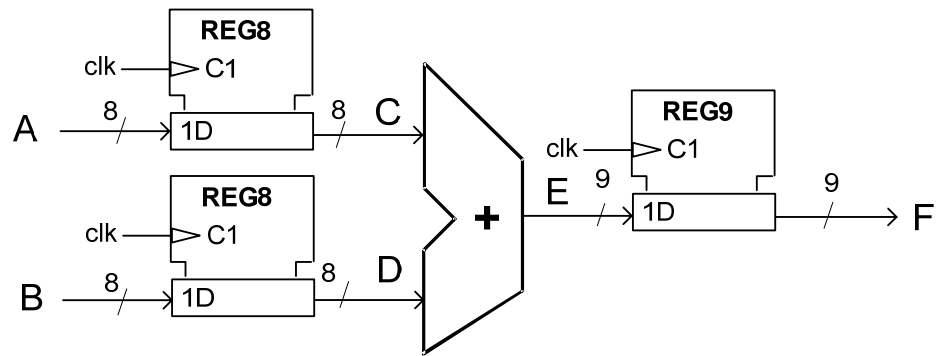
Tentti: 7.12.2010

Tehtävä 1

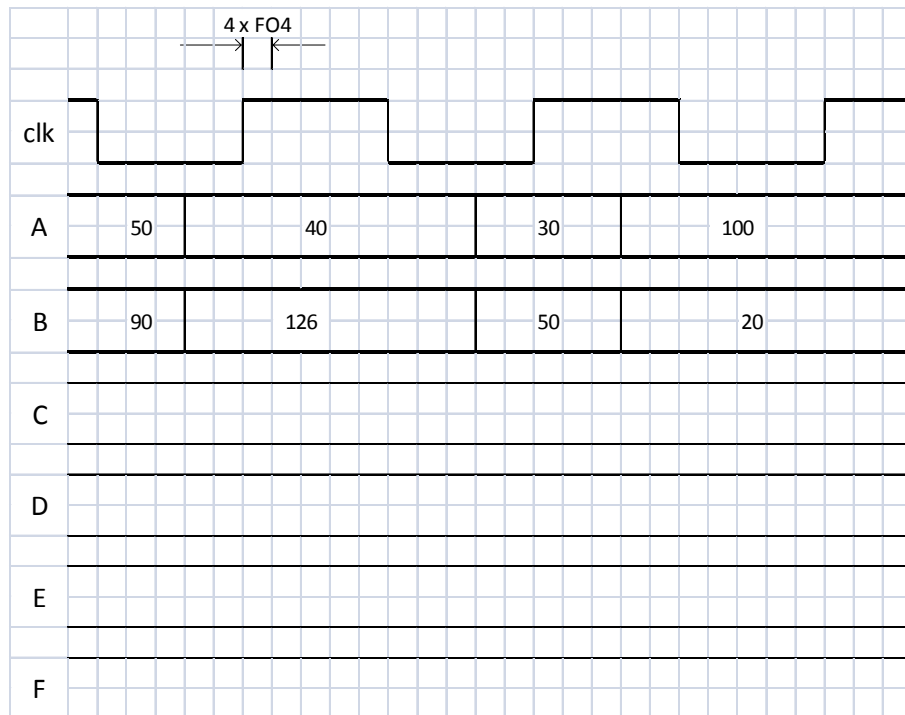
Kuvassa 1-1 on esitetty synkroninen RTL-datapolku, joka muodostuu kolmesta rekisteristä ja summaimesta. Kuvassa 1-2 on kuvan 1 datapolkuun liittyvä simulaattorin aaltomuotonäkymä (*Wave Form View*), jossa on näkyvissä synkroninen kello-signaali (clk) ja signaalipisteiden A ja B aaltomuodot. Kuvan numerot ovat väylien binäärilukuja vastaavia desimaalilukuja.

Rekisterien toteutuksessa käytettyjen D-kiikkujen asettumisaika (setup time), $\tau_{\text{setup}} = 4 \times \text{FO4}$, ja viive $\tau_{\text{diff}} = 8 \times \text{FO4}$. Summain suorittaa tehtävänsä ajassa, joka on 20 FO4 -viiveen pituinen.

- a) Suorita kuvan 1-1 arkkitehtuurin niin sanottu ”pöytäsimulointi” täydentämällä aaltomuotoikkunaan signaalipisteiden C, D, E ja F aaltomuodot. Tässä tapauksessa aaltomuodot voidaan ilmoittaa signaalipisteen bittien muodostamaa binäärilukua vastaavina desimaalilukuina.
- b) Selitä käsitteet FO4-viive, latenssi, etenemisviive, kriittinen polku ja maksimikellotaajuus.
- c) Jos kuvan 1-1 arkkitehtuuri toteutetaan valmistusteknologialla, jonka FO4-viive on 100 ps, niin paljonko silloin on latenssi, kriittisen polun pituus (*critical path*) ja maksimikellotaajuus. Montako yhteenlaskua sekunnissa tehdään simulointituloksen arvoilla (kuva 1-2)? Miten tilanne muuttuu, jos käytetään maksimikellotaajuutta?



Kuva 1-1. Tehtävän 1 datapolku.

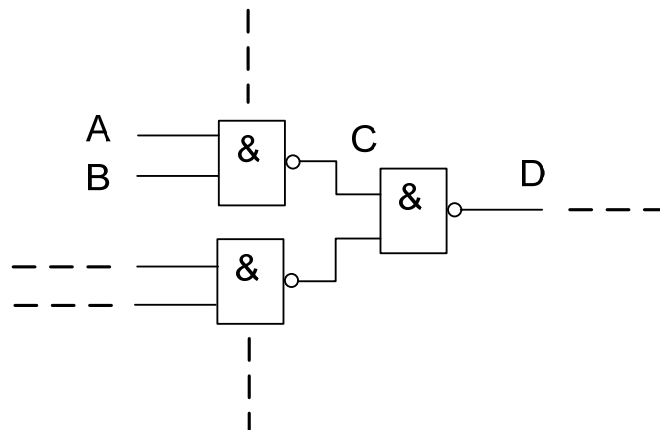


Kuva 1-2. Tehtävän 1 simuloinnin aaltomuotonäkymä (kuvassa yhtä ruutua aikatasossa vastaa aika, joka on 4 FO4-viiveen pituinen).

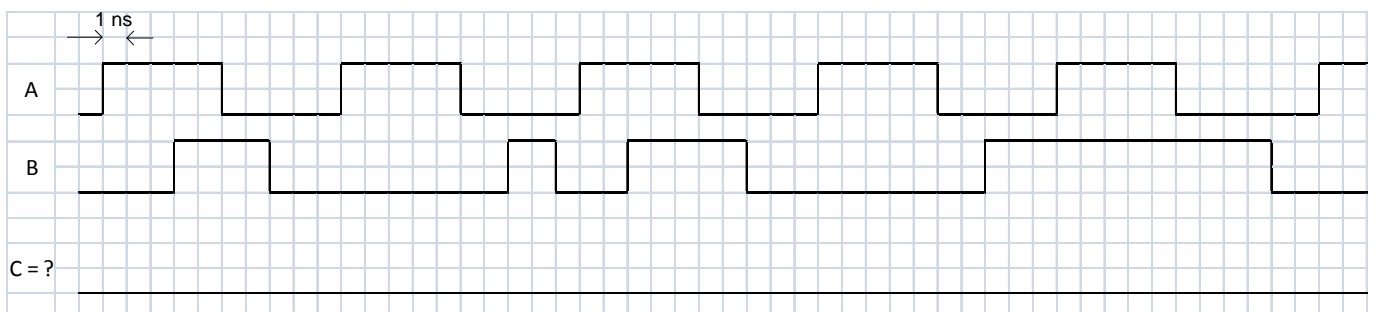
Tehtävä 2

- Selosta miten CMOS-logiikan dynaaminen ja staattinen tehonkulutus tapahtuu?
- Mitä tietoja tarvitset voidaksesi arvioida CMOS-logiikan tehonkulutusta rekisteri-siirtotason (RTL) arkkitehtuurikuvauksen perusteella?
- Kuvassa 2-1 on näkyvissä osa laajemmasta logiikkatason kaaviosta.
 - Paljonko signaalipisteessä C kuluu **energiaa** sinä aikana kun kuormituskapasitanssin jännite laskee loogiseen nollatilaan (kapasitanssi purkautuu) ja nousee taas takaisin loogiseen ykköstilaan (kapasitanssi latautuu).
 - Paljonko on kokonais**energian**kulutus pisteessä C kuvassa 2-2 kuvatun tarkastelujakson aikana (taulukossa yksi ruutu vastaa aikatasossa 1 ns:n pituista aikaa).
- Laske keskimääräinen **tehon**kulutus kuvan 2-1 signaalipisteessä C kuvassa 2-2 esitetyn tarkastelujakson aikana.

Kuvan 2-1 porttien tulokapasitanssi $C_{in} = 1 \text{ fF} = 10^{-15} \text{ F}$ ja käyttöjännite on 1 V.



Kuva 2-1. Osa laajemmasta logiikkakaaviosta. Tehtävässä kysytään energian ja tehonkulutusta signaalipisteessä C.



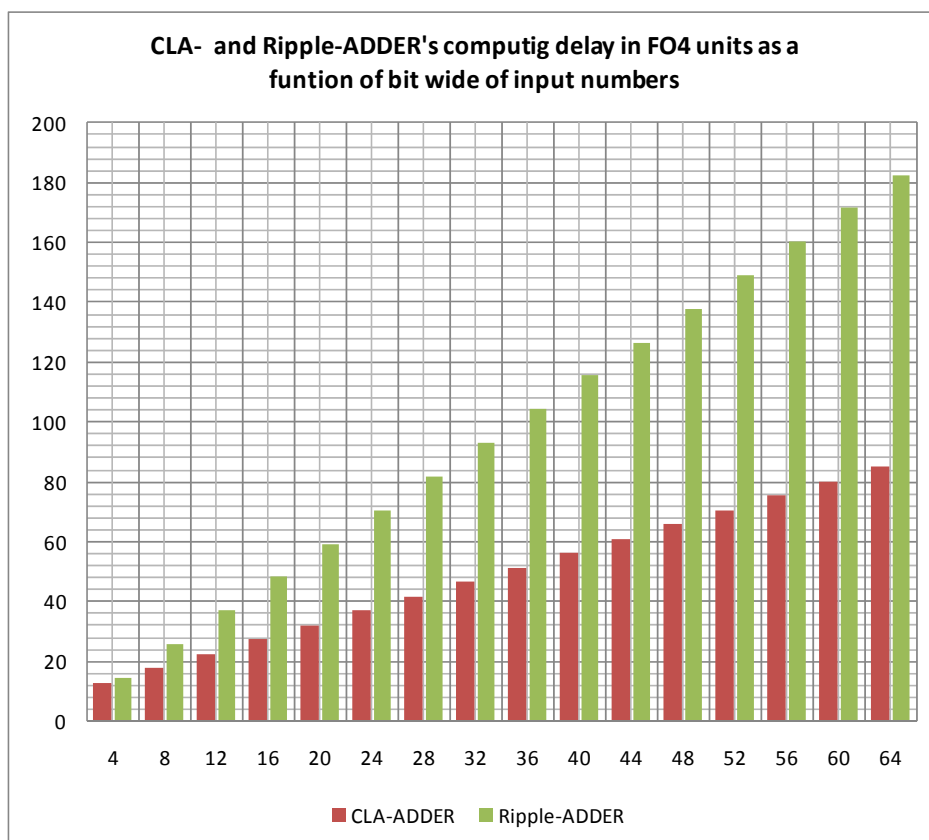
Kuva 2-2 Kuvan 2-1 logiikan herätteet signaalipisteissä A ja B. (Yksi ruutu = 1 ns).

Tehtävä 3

Suunnittele RTL-arkkitehtuuri, jonka tehtävänä on laskea yhteen kahdeksan 2:n komplementti-esitys muodossa olevaa binäärilukua: A (4 bittiä), B (6 bittiä), C (8 bittiä), D (10 bittiä), E (12 bittiä), F (13 bittiä), G (14 bittiä) ja H (16 bittiä).

- Minkälainen arkkitehtuuri on, jos sen pitää suoriutua tehtävästä mahdollisimman nopeasti.
- Minkälainen arkkitehtuuri on, jos sen pitää olla mahdollisimman pieni eli sen toteutuksessa tarvitaan logiikkaa mahdollisimman vähän.
- Arvioi montako kokosummainta (FA = Full Adder) suunnittelemissi a- ja b-kohtien arkkitehtuurien toteutuksessa tarvitaan.
- Arvioi kuinka kauan lukujen summaus kestää a- ja b-kohtien arkkitehtuureissa. Käytä laskentanopeuden arvioinnissa apuna kuvan 3-1 taulukkoa, jossa on CLA-ADDER:in ja Ripple-ADDER:n nopeus FO4-yksikköinä 4 bitin välein 4:stä 64:ään.

Kuvan 3-1 lukuohje: Mikäli summaimen bittimäärä ei ole 4:llä jaollinen, niin katso sen nopeus lähinnä isomman summaimen kohdalta (esim. 5 bitin summaimen nopeus on tässä sama kuin 8 bitin summaimen). Lue viivearvoja 4 FO4:n tarkkuudella.



Kuva 3-1. Carry Look-Ahead (CLA) ja "ripple carry" summaimien nopeus FO4-yksikköinä.

Tehtävä 4

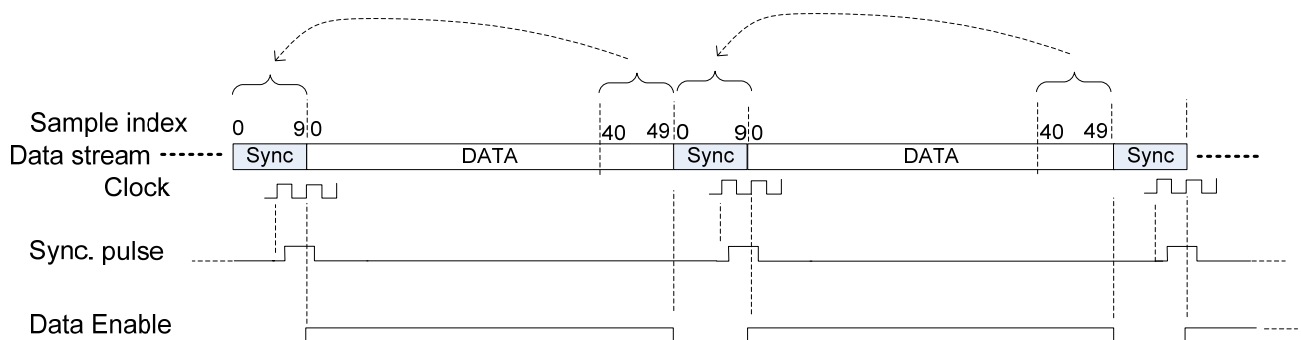
Suunnittele synkronointilogiikka, joka ilmaisee datavirrasta hyötydatajaksojen alkuhetket eli kellosignaalin ne nousevat reunat, joilla hyötydatajakson ensimmäinen datanäyte esiintyy.

Datavirta, johon suunniteltavan logiikan on synkronoiduttava, muodostuu 8-bittisistä binäärivektoreista, joita saadaan käsiteltäväksi kellosignaalin nousevan reunan määrittäminä ajanhetkinä.

Jatkuvassa datavirrassa on toistuvasti 50 näytettä hyötydataa ja 10 näytettä synkronointijaksoa. Synkronointijakso on aina kopio sitä seuraavan datajakson 10 viimeisestä näytteestä (Kts. kuva 4-1).

Esitä suunnitelmasi RTL-arkkitehtuurin muodossa niin, että se sisältää datapolun (*data path*) ja ohjausosan (*control part*) kuvauksen.

Bonustehtävä: Synkronointilogiikan VHDL- tai Verilog-kielisestä mallista voit saada lisäpisteitä maksimissaan yhden tehtävän verran.



Kuva 4-1. Datavirran kehysrakente ja synkronoinnissa tarvittavien signaalien ajoituskaavio.