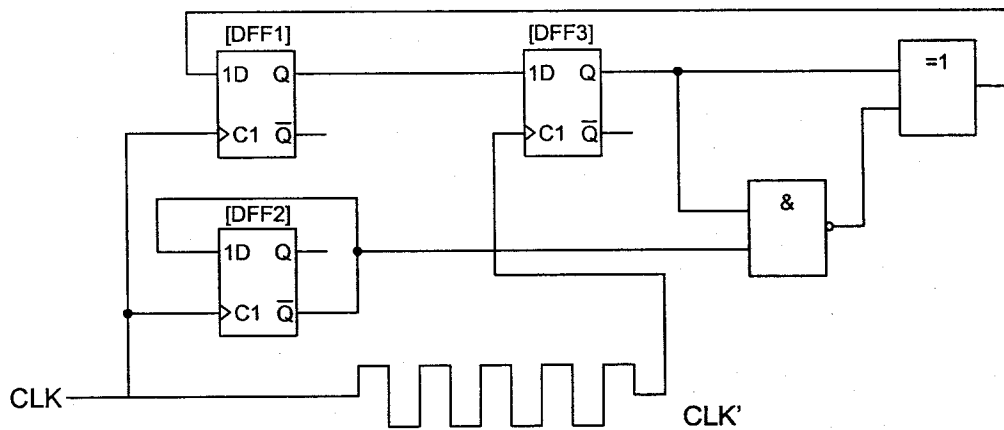


TEHTÄVÄ 1.

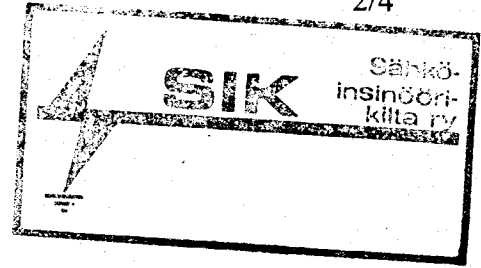
Kuvassa 1 on esitetty synkroninen digitaalipiiri, jossa kiikkujen ajoitusparametrit ovat $t_{DFF} = 5$ ns, $t_{setup} = 1,5$ ns ja $t_{hold} = 1$ ns sekä NAND- ja XOR-porttien viiveet $t_{nand} = t_{xor} = 2$ ns.

a) Piiriä kellotetaan 50 MHz:n taajuudella ($T_{clk} = 20$ ns). Kiikulle DFF3 menevä kellosignaali viivästyy johdotuksen aiheuttaman kuluaikaviiveen takia. Kuinka suuri tämä kellopoikkeama t_{skew} saa maksimissaan olla, jotta piiri toimisi vielä oikein? Perustelee vastauksesi joko ajoituskaaviolla tai laskulla. (4p)

b) Jos tiedämme, että kiikun DFF3 kellosignaali on viivästynyt tarkalleen 2 ns, niin mikä on lyhin kellojakso, jolla piiri toimii luotettavasti? (2p)



Kuva 1. Digitaalinen piiri.



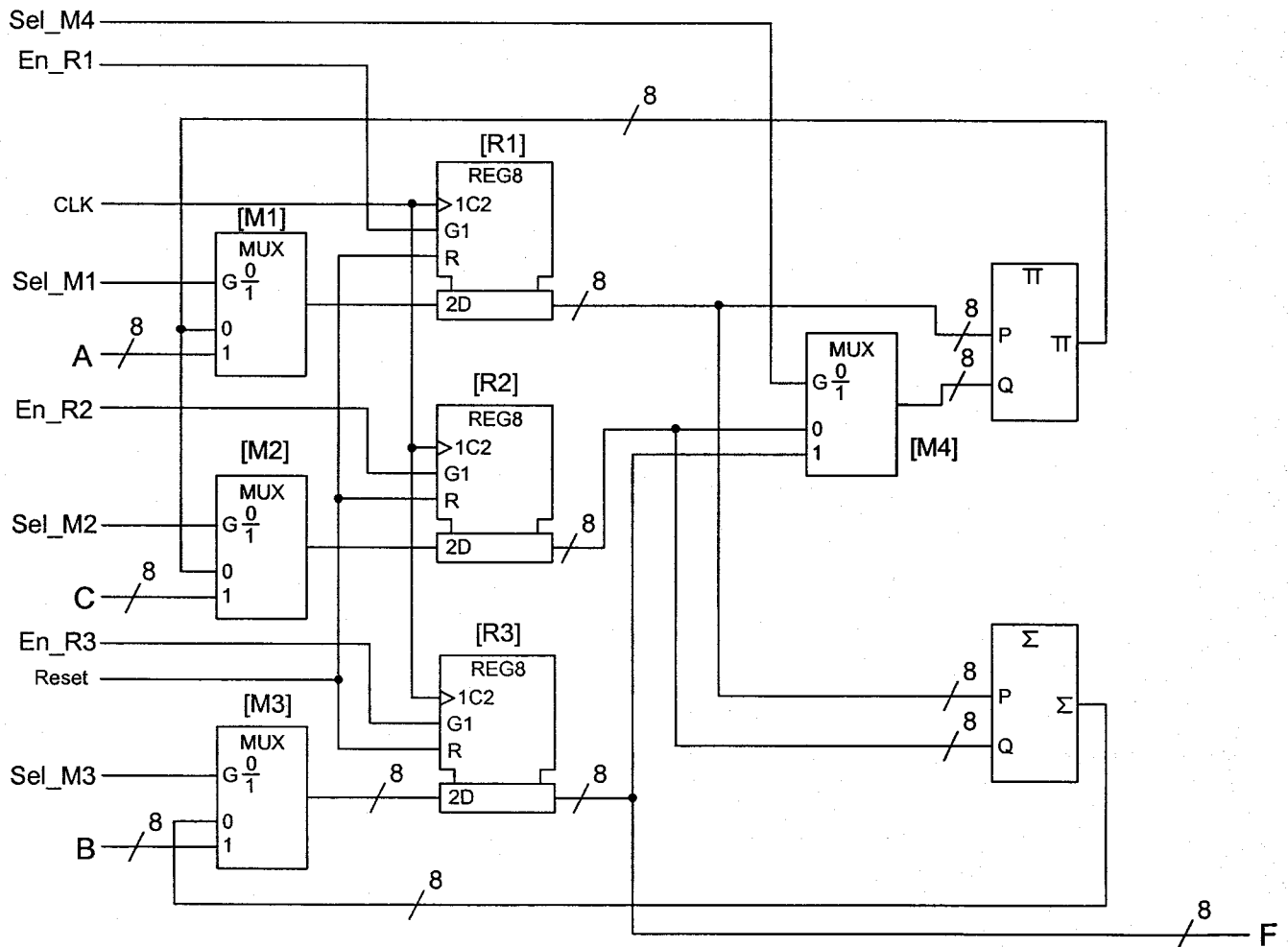
TEHTÄVÄ 2.

Kuvassa 2 on esitetty digitaaliarkkitehtuuri tietoväylästä, jolla voidaan laskea algoritmi $F=AB+ABC$ neljässä kellojaksossa. Kuvaan on nimetty myös tietoväylän ohjaussignaalit.

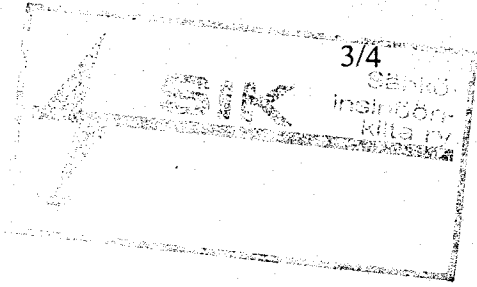
a) Suunnittele digitaaliarkkitehtuuri ohjausosasta, jolla voidaan generoida tarvittavat ohjaussignaalit tietoväylälle algoritmin $F=AB+ABC$ laskemiseen. (3p)

b) Esitä algoritmin $F=AB+ABC$ laskemiseen tarvittava ohjaussekvenssi, eli tietoväylän ohjausvektori kellojaksokoittain. (3p)

Vihje: ensimmäisellä kellojaksolla muuttujat A, B ja C talletetaan rekistereihin.



Kuva 2. Tehtävän 2 tietoväylä.

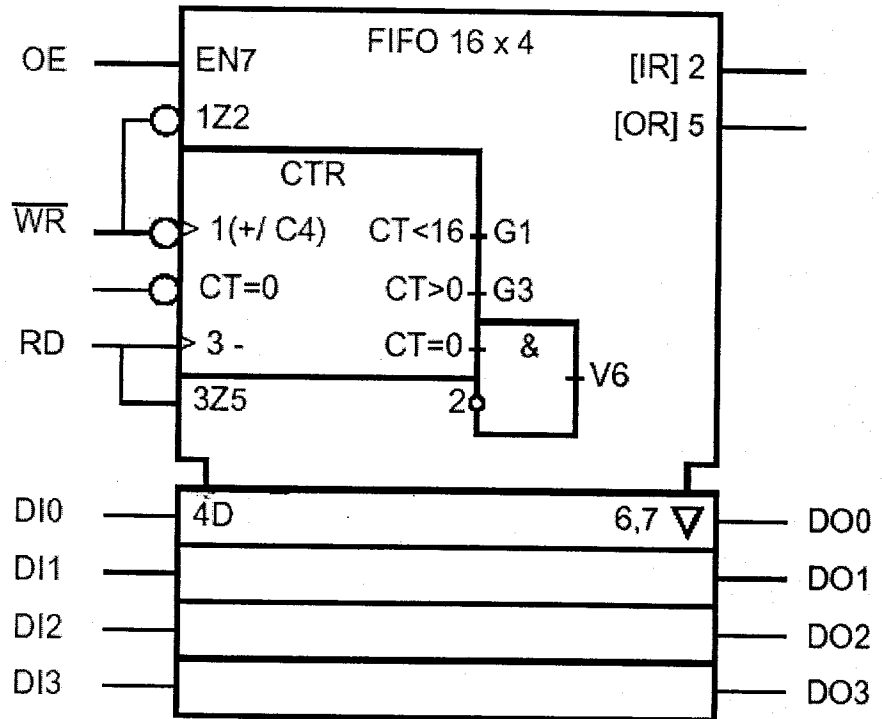


TEHTÄVÄ 3.

Tehtävänäsä on tutustua alla olevaan kurssimonisteen kuvaan 6.23 ja siihen liittyvään selittävään tekstiin ja piirtää sen perusteella piirrosmerkin kuvaama arkkitehtuuri uudestaan hajotettuna erillisiksi komponenteiksi ja niiden väliseksi kytkennäksi, joka muodostuu laskurista (CTR4), 16 neljän bitin rekisteristä ja riippuvuusmerkintöjä vastaavista logiikkaporteista.

Ohje: Riippuvuusmerkintä **G** vastaa JA-porttia, **V** TAI-porttia ja **Z** suoraa kytkentää. **EN**-riippuvuutta käytetään tässä kolmitilälähdön sallintatulona (JA-riippuvuus).

”Kuvan 6.23 piirrosmerkki havainnollistaa erään FIFO-muistipiirin toimintaa. Tässä piirrosmerkissä FIFO:n toimintaa kuvataan laskurilohkolla (CTR). Kirjoitettaessa laskuria kasvatetaan ja luettaessa sitä vähennetään. Kirjoitus on sallittu vain kun laskurin sisältö on pienempi kuin 16, mikä on FIFO:n kapasiteetti. Vastaavasti luku on sallittu, kun laskurin sisältö on suurempi kuin 0, mikä osoittaa että FIFO: on kirjoitettu dataa. Piirissä on myös laajennuslähdtöt [IR] ja [OR], jotka kertovat, onko piirille luku tai kirjoitus mahdollista.”



Kuva 6.23. FIFO-muistin piirrosmerkki.

TEHTÄVÄ 4

Piirrä viisitappisen ($K = 5$) FIR-suodattimen RT-tason arkkitehtuurista kaksi versiota, joista ensimmäisen (a) logiikkatason toteutukseen tarvitaan mahdollisimman vähän portteja (transistoreita) ja toinen (b) laskee suodatettuja näytteitä mahdollisimman nopeasti.

Ohje:

FIR (Finite Impulse Response) –algoritmin matemaattinen esitysmuoto on

$$y_n = \sum_{k=0}^{K-1} C_k x_{n-k} ,$$

jossa näytteen aikaindeksi, y_n = suodatettu näyte hetkellä n , x on signaalinäyte, k on kertoimen indeksi, C_k on k :s suodatuskerroin ja K on suodattimen tappien lukumäärä.

Esimerkiksi 256 tappisen FIR-suodatinalgoritmin toimintaa voidaan kuvata seuraavan C-koodimallin avulla:

```
float FIR ( float x[], float c[])  
{int i ; int K = 256;  
  float sum = 0. ;  
  for (i = 0; i < K; i++)  
  {  
    sum += x[i] * c[i] ;  
  }  
  return sum;  
}
```