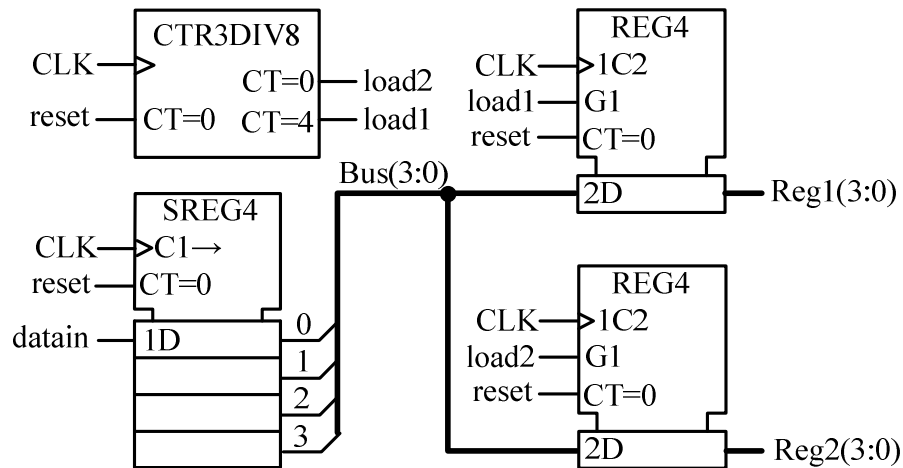


Tehtävä 1

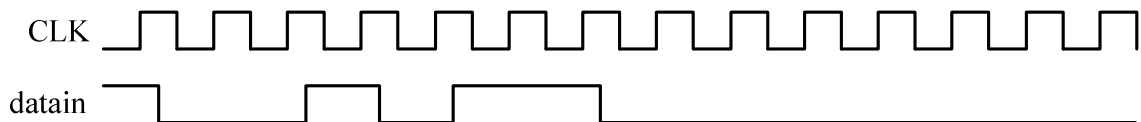
a) Selitä lyhyesti oheisen RTL-tason logiikan toiminta.

b) Piirrä ajoituskaavio, jossa tulosignaali datain saa kuvan mukaisesti arvot 10010110000000 kellon CLK tahdissa. Esitä ajoituskaaviossa tulosignaalin datain ja CLK lisäksi lähtösignaalien Bus(3:0), Reg1(3:0), Reg2(3:0), load1 ja load2 käyttäytymisen.

Kaikkien kiikkujen/rekistereiden alkutilaksi on alustettu signaalilla reset looginen nolla, ja RTL on vapautettu resetistä ja on valmiina käsittelemään dataa. Väylät on piirretty paksummalla viivalla kuin yksittäiset signaalit.



Tehtävän 1 RTL-tason logiikkakaavio



Tehtävän 1 signaalien CLK ja datain ajoituskaavio

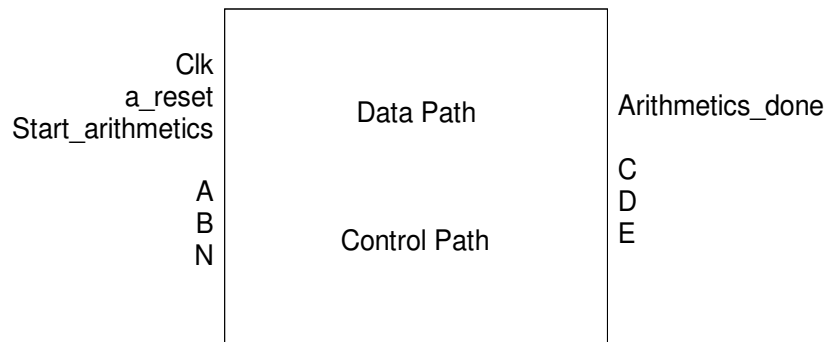
Tentti: 8.12.2009

Tehtävä 2

Tehtävänäsi on suunnitella datapolku ja kontrolliosia, jotka toteuttavat seuraavat toiminnot:

$$C = A + B, \quad D = A - B, \quad E = A * N.$$

Datapolussa saa käyttää vain yhtä summainta, jonka kautta kaikki aritmeettiset operaatiot suoritetaan. Kun "start_arithmetics" käytetään ylhäällä ainakin yksi kellojakso, laskee alla oleva lohko vaadittavat tulokset lähtöihin. Kun laskenta on valmis, nostaa lohko "arithmetics_done" lähdön ylös, kunnes uusi iteraatio alkaa. A_reset on nolla-aktiivinen. A ja B ovat 24-bittisiä kahden komplementtilukuja ja N on kokonaisluku välillä 1-15. Mitoita tarvittavat bittileveydet lähdöille C, D ja E. Pidä datapolku ja kontrolliosia erillään opetetulla tavalla. Piirrä tarvittavat RTL:t ja tilakaaviot.



Tehtävä 3

- Luettele digilogiikan synkronisuuden tunnusmerkit.
- Mitä ominaisuuksia digilogiikan suunnitelmasta (arkkitehtuurikaavio, VHDL malli) pitää löytyä, jotta sen voitaisiin sanoa olevan rekisterisiirtotasolla (RTL = *Register Transfer Level*)?
- Mitä CMOS-teknologialla toteutetun integroidun piirin piirikuvioinnin viivanleveydellä tarkoitetaan?
- Mitä integroidun digitaalipiirin piipinta-alan arvioinnissa käytetty *Cell Factor* –tarkoittaa?
- Selitä minkälainen FPGA-piirin hakutaulukko on **loogiselta** rakenteeltaan?

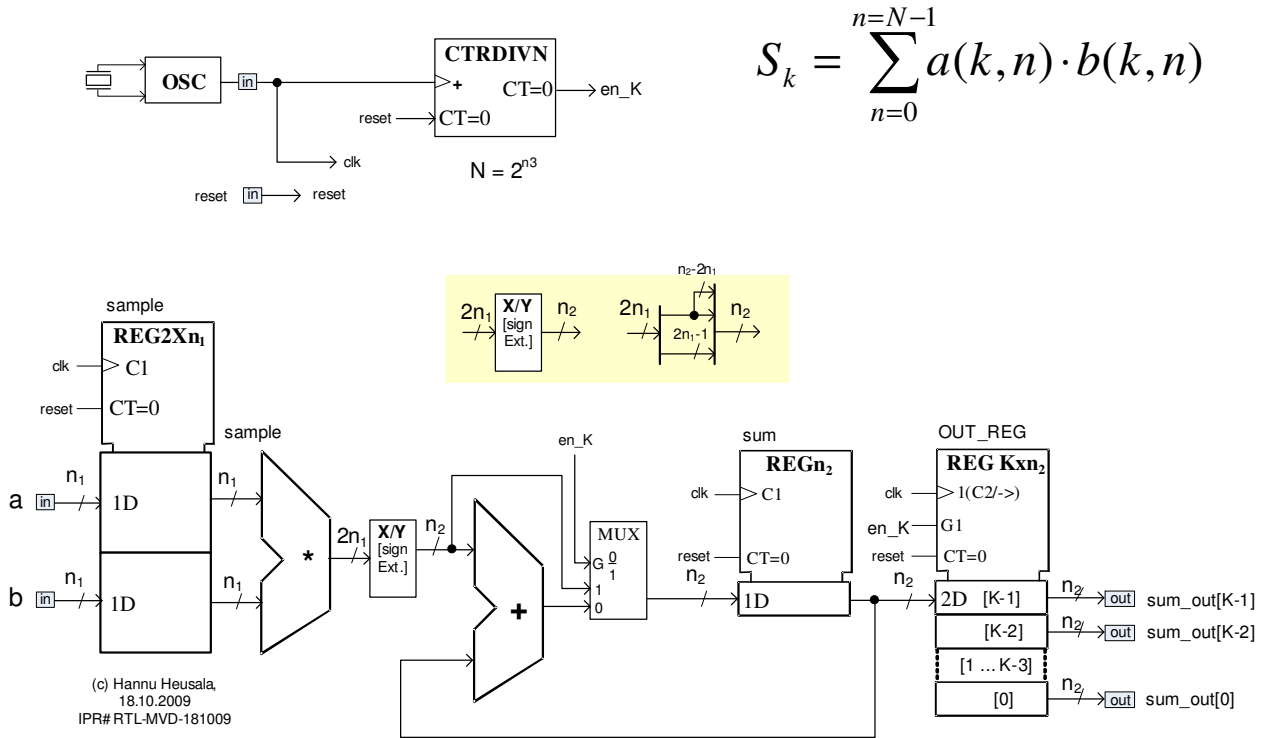
Tehtävä 4

Liitteessä 1 näet RTL-arkkitehtuurin ja liitteessä 2 sitä vastaavan VHDL-mallin. Kysymyksessä on Multiply-Cumulate-Add (MAC) operaation toteuttava digilogiikka.

Arkkitehtuurista saadaan sanan- ja laskentasekvenssin pituuksiltaan erilaisia versioita muuttamalla parametrien (n_1 , n_2 , ja K) arvoja. Oletetaan, että $K = N/2$.

- Päättele arkkitehtuurin ja VHDL-mallin perusteella mikä on parametrin n_2 optimaalinen arvo, jos parametrien n_1 ja n_3 arvot on annettu.
- Oletetaan, että liitteissä 1 ja 2 kuvattu arkkitehtuuri konfiguroidaan FPGA-piiriin siten, että kertojana käytetään valmista 18x18-bitin lohkokertojaa ja summaimena 4-tuloisilla hakutaulukoilla (4-LUT) toteutettua *ripple-carry* –tyyppistä rakennetta. Jos parametri $n_1 = 8$, n_2 :lla on optimaalinen arvonsa ja $n_3 = 4$, niin montako D-kiikkua ja hakutaulukkoa (4-LUT) toteuttamiseen tarvitaan?
- Minkä komponenttien kautta arkkitehtuurin kriittinen polku kulkee? Mitkä parametrit (n_1 , n_2 , n_3 ja K) vaikuttavat kriittisen polun pituuteen ja miten?
- Kirjoita parametrinen (n_1 , n_2 , n_3 ja K) avulla kaksi matemaattista lauseketta: toinen D-kiikkujen ja toinen hakutaulukkojen lukumäärän arvioimista varten.

LIITE 1: Tehtävään 4 liittyvä RTL-arkkitehtuuri:



Tentti: 8.12.2009

LIITE 2: Tehtävän 4 RTL-arkkitehtuuria vastaava VHDL-malli:

```

library ieee; use ieee.std_logic_1164.all; use ieee.std_logic_signed.all;
--
-- (c) Hannu Heusala 18-10-2009
-- MAC design for DT courses
-- Verification: 18-10-2009, RTL-VHDL-hhh-181009
--
entity MAC is
  generic(n1:natural:=8; n2:natural:=20;n3:natural:=4;K:natural:=8);
  port(clk,reset:in std_logic;
  sample_in_a,sample_in_b:in std_logic_vector(n1-1 downto 0);
  sum_out:out std_logic_vector(n2-1 downto 0));
end entity MAC;
--
architecture RTL1 of MAC is
  type OUT_REG_type is array(K-1 downto 0) of std_logic_vector(n2-1 downto 0);
  signal sample_a, sample_b:std_logic_vector(n1-1 downto 0);
  signal product: std_logic_vector(2*n1-1 downto 0);
  signal sum,P:std_logic_vector(n2-1 downto 0);
  signal OUT_REG:OUT_REG_type;
  signal CTR: std_logic_vector(n3-1 downto 0);
  --
begin
  sync:process(clk,reset)
    variable en_K:std_logic:='0';
  begin
    if CTR=0 then en_K:='1'; else en_K:='0';end if;
    if reset='1' then
      sample_a<=(others=>'0');sample_b<=(others=>'0');
      sum<=(others=>'0');OUT_REG<=(others=>(others=>'0'));
      CTR <= (others=>'0');
    elsif rising_edge(clk)then
      sample_a<=sample_in_a; sample_b<=sample_in_b;
      sum<=P+sum; CTR<=CTR+'1';
      if en_K='1' then
        sum<=P;
        OUT_REG(K-1)<=sum;
        OUT_REG(K-2 downto 0)<=OUT_REG(K-1 downto 1);
      else OUT_REG<=OUT_REG;
      end if;
    end if;
  end process sync;
  --
  product<=sample_a*sample_b;
  --
  --sign extend:
  P(n2-1 downto 2*n1-1)<=(others => product(2*n1-1));
  P(2*n1-2 downto 0)<= product(2*n1-2 downto 0) ;
  --
  sum_out<=OUT_REG(0);
  --
end architecture RTL1;

```