

TKT-1212 Digitaalijärjestelmien toteutus, tentti ma 19.05.2009  
Laatinut Erno Salminen. Laskinta saa käyttää.

Jos suoritat tenttiä, vastaa kaikkiin kysymyksiin.

Jos suoritat 2. välikokeen, astaa vain kysymyksiin 1-3.

Tekstin lisäksi suosi kaavoja, kuvia, taulukoita ja esimerkkejä vastauksissasi. Merkitse nimesi ja op.numerosi jokaiseen palauttamaasi paperiin.

Logiikkakuvissa voit käyttää perusporttien (AND, OR...) lisäksi kiikkuja, multipleksereitä, vertailijoita, sekä yhteen-, vähennys- ja kertolaskuyksiköitä. Nimeä kaikki signaalit ja merkitse niiden leveys selkeästi. Jos tehtävänanto on väljä tai epäselvä, kerro tarkasti mitä oletuksia teit.

1. Synkronointi

- a) Selitä metastabiilisuus. Mitä tarkoittaa, milloin esiintyy, mitä siitä seuraa? (2p)
- b) Piirrä nk. "kahden kiikun synkronoija" (engl. *two-DFF synchronizer*) ja selitä sen toiminta. (2p)
- c) Toimiiko synkronoija, jos käytätkin vain 1 tai peräti 3 kiikkua? (1p)
- d) Mitkä kaikki signaalit pitää synkronoida? (1p)

2. Kuvaile nykyaikaisen FPGA-piirin rakenne. (3p)

3. Analysoi seur. sivun VHDL-koodia.

- a) Esitä koodista syntesoituva logiikka. (3p)
- b) Miten syntesoitu logiikka toimii eri tavalla kuin RTL-simulaatio? (3p)

Välikoe päättyy tähän.

4. Analysoi seur. sivun VHDL-koodia.

- a) Mitä tyyppiä prosessit ovat? (2p)
- b) Mitä virheitä ja epäilyttäviä kohtia koodissa on? Siis muita kuin kohdassa 3b mainitut (3p)
- c) Täydennä ajoitukaavio kuten simulaattori koodin tulkitsee. Kellojakson pituus on 10 ns. (5p)

5. Selitä seuraavat käsitteet (á 1p)

- a) Signaaliajuri (*signal driver*)
- b) Herkkyyslista (*sensitivity list*)
- c) Arkkitehtuurityylien *RTL* ja *structural* ero.
- d) "Maaginen numero" (*magic number*).
- e) Kombinatorinen silmukka (*combinatorial loop*)

# TKT-1212 Digitaalijärjestelmien toteutus, tentti ma 19.05.2009

## Tehtävien 3 ja 4 koodi

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
use ieee.numeric_std.all;
```

```
entity valikoe2 is
```

```
  generic (
```

```
    data_width_g : integer := 3);
```

```
  port (
```

```
    clk      : in std_logic;
```

```
    rst_n    : in std_logic;
```

```
    a_in     : in std_logic;
```

```
    b_in     : in std_logic;
```

```
    data_in  : in std_logic_vector (data_width_g-1 downto 0);
```

```
    data0_out : out std_logic_vector (data_width_g-1 downto 0);
```

```
    data1_out : out std_logic_vector (data_width_g-1 downto 0)
```

```
  );
```

```
end valikoe2;
```

```
architecture rtl of valikoe2 is
```

```
  signal c      : std_logic;
```

```
  signal tmp    : std_logic_vector (data_width_g-1 downto 0);
```

```
  signal tmp_r  : std_logic_vector (data_width_g-1 downto 0);
```

```
begin
```

```
  king : process (a_in, b_in, c)
```

```
  begin -- process king
```

```
    c <= a_in or b_in;
```

```
    if c = '1' then
```

```
      tmp <= data_in;
```

```
    else
```

```
      tmp <= (others => 'Z');
```

```
    end if;
```

```
end process king;
```

```
  data0_out <= tmp;
```

```
  hanneman : process (clk, rst_n, tmp)
```

```
  begin -- process hanneman
```

```
    if rst_n = '0' then -- asynchronous reset (active low)
```

```
      elsif clk'event and clk = '1' then -- rising clock edge
```

```
        tmp_r <= "110";
```

```
        tmp_r <= tmp;
```

```
      end if;
```

```
    end process hanneman;
```

```
    data1_out <= tmp_r after 4 ns;
```

```
end rtl;
```

NIMI:

OP.NUMERO:

Teht. 4

