

Vermeld op elk blad: naam en voornaam. Alle pagina's nummeren.

De voorbereidingstijd voor de eerste twee vragen samen is maximum 1 uur, voor de derde vraag 1,5 uur.

De vierde vraag wordt schriftelijk opgelost en afgegeven ten laatste om 13 u.

1. Wat is een Schmitt-trigger ingang en waarom zou men die willen gebruiken?
2. Wat is 'pipelining' en welke impact heeft dit op het ASM-schema?
3. Ontwerp een FSMD die $3e^x$ benadert zoals dit op de achterkant van dit blad in VHDL beschreven is. Bij de realisatie mag je enkel bouwblokken gebruiken waarvan de hardware implementatie in de les bestudeerd is.

De ingang id wordt voorgesteld in 1-complement terwijl de uitgang od in 2-complement voorgesteld wordt. Alle real getallen worden in hardware voorgesteld als getallen met een vaste komma. Voor de ingang id en de uitgang od worden 8 bits na de komma gebruikt. Verder worden voor alle getallen zoveel bits gebruikt als nodig om de uitgang met de gewenste nauwkeurigheid te berekenen.

- (a) Teken een ASM-schema voor deze schakeling, die de vereisten qua tijdsgedrag respecteert. Probeer zo weinig mogelijk toestanden te gebruiken. Indien één enkel ASM-schema te ingewikkeld is mag je dit opsplitsen in meerdere ASM-schema's.
 - (b) Ontwerp het datapad tot op RTL-niveau. Minimalisering is niet expliciet nodig, maar probeer wel zo weinig mogelijk hardware te gebruiken, zonder evenwel nog het ASM-schema te wijzigen. Vergeet ook niet het aantal bits bij iedere verbinding te vermelden.
 - (c) Beschrijf het controlegedeelte met een toestandsdiagram. Als er meerdere ASM-schema's gebruikt worden volstaat het om enkel het diagram met de meeste toestanden te beschrijven.
4. Maak de goedkoopst mogelijke IC realisatie van onderstaande FSM waarbij je enkel gebruik mag maken van T-flipflops en NOR-poorten met 2 ingangen. Maak gebruik van Karnaugh-kaarten om alle functies te bepalen.

ab	01	10	11
S_0	$S_3/10$	$S_0/01$	$S_1/10$
S_1	$S_2/10$	$S_4/01$	$S_0/10$
S_2	$S_1/10$	$S_0/01$	$S_4/10$
S_3	$S_4/10$	$S_3/10$	$S_3/01$
S_4	$S_3/10$	$S_4/01$	$S_2/10$

```

entity fsmd is
  port (clk, rst : in bit; LD : in integer range -1 to 4;
        id : in real range -1.0 to 1.0;
        od : out real; oa : out bit);
end entity fsmd;

architecture behav of fsmd is
  signal n, rsti : natural;
  signal odi, x : real;
begin
  p1: process is
    variable t, y : real;
  begin
    if n > 0 then
      oa <= '0';
      y := 1.0; if n < 3 or x = 0.0 then y := 3.0; end if;
      for i in n downto 1 loop
        exit when rsti > 0;
        t := x*y;
        if i and 1 = 0 then t := t/i; end if;
        y := t + 1.0;
        if i > 2 then y := y + 2.0; end if;
        wait until clk = '1';
      end loop;
      if rsti = 0 then
        oa <= '1'; odi <= y;
      end if;
      n <= 0;
    end if;
    wait until clk = '1';
  end process p1;

  p2: process (clk, rst, odi) is
    variable ig, ni : natural;
    variable xi : real;
  begin
    if rst = '0' then
      n <= 0; rsti <= 1; oa <= '0'; ig := 0;
    elsif clk'event and clk = '1' then
      rsti <= 0;
      if n = 0 then
        if LD < 0 then xi := id; ig := ig or 1;
        elsif LD > 0 then ni := LD; ig := ig or 2;
        end if;
        if ig = 3 then
          x <= xi; n <= ni; ig := 0;
        end if;
      end if;
    end if;
    od <= odi;
  end process p2;
end architecture behav;

```