

# Fasit INF3430/4430 Eksamen H-2008

## Oppgave 1-14

Oppgave	A	B	C	D
1		X		X
2		X		
3	X		X	X
4		X		X
5	X			
6			X	X
7	X	X		
8		X	X	X
9	X		X	X
10	X		X	
11	X			X
12	X	X		X
13		X		
14		X		

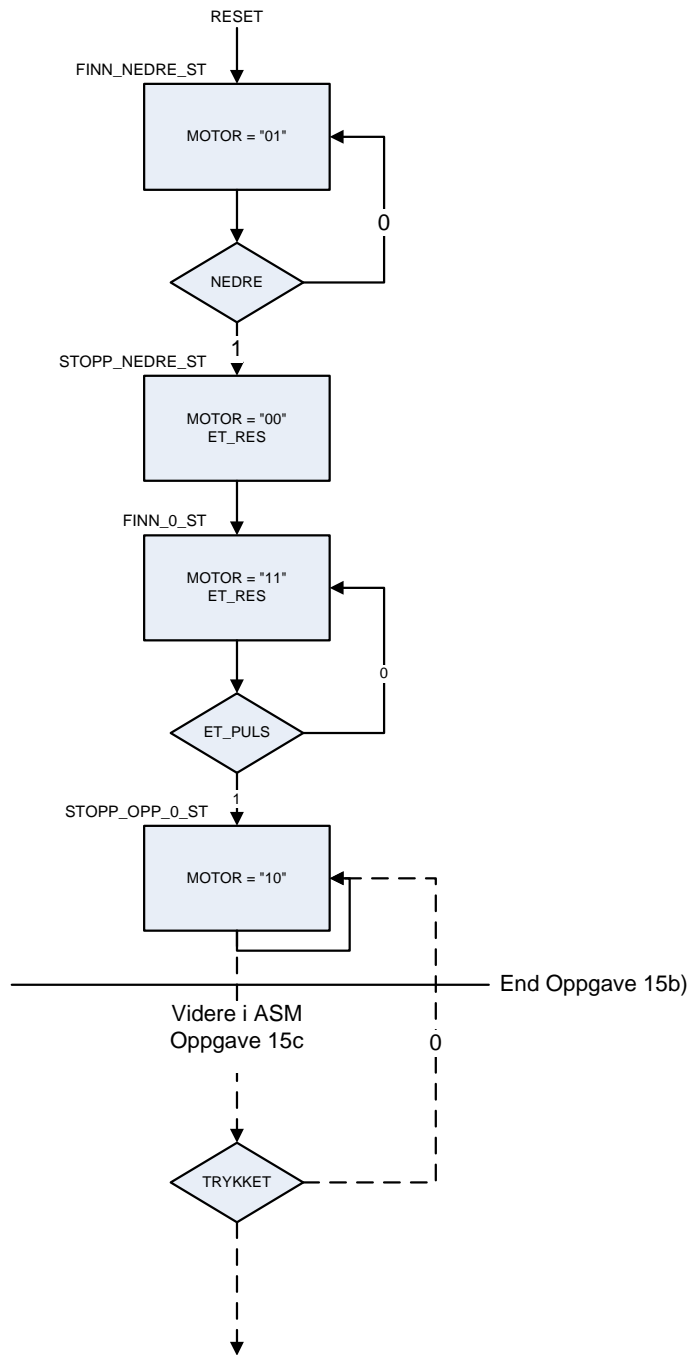
## Oppgave 15

a)

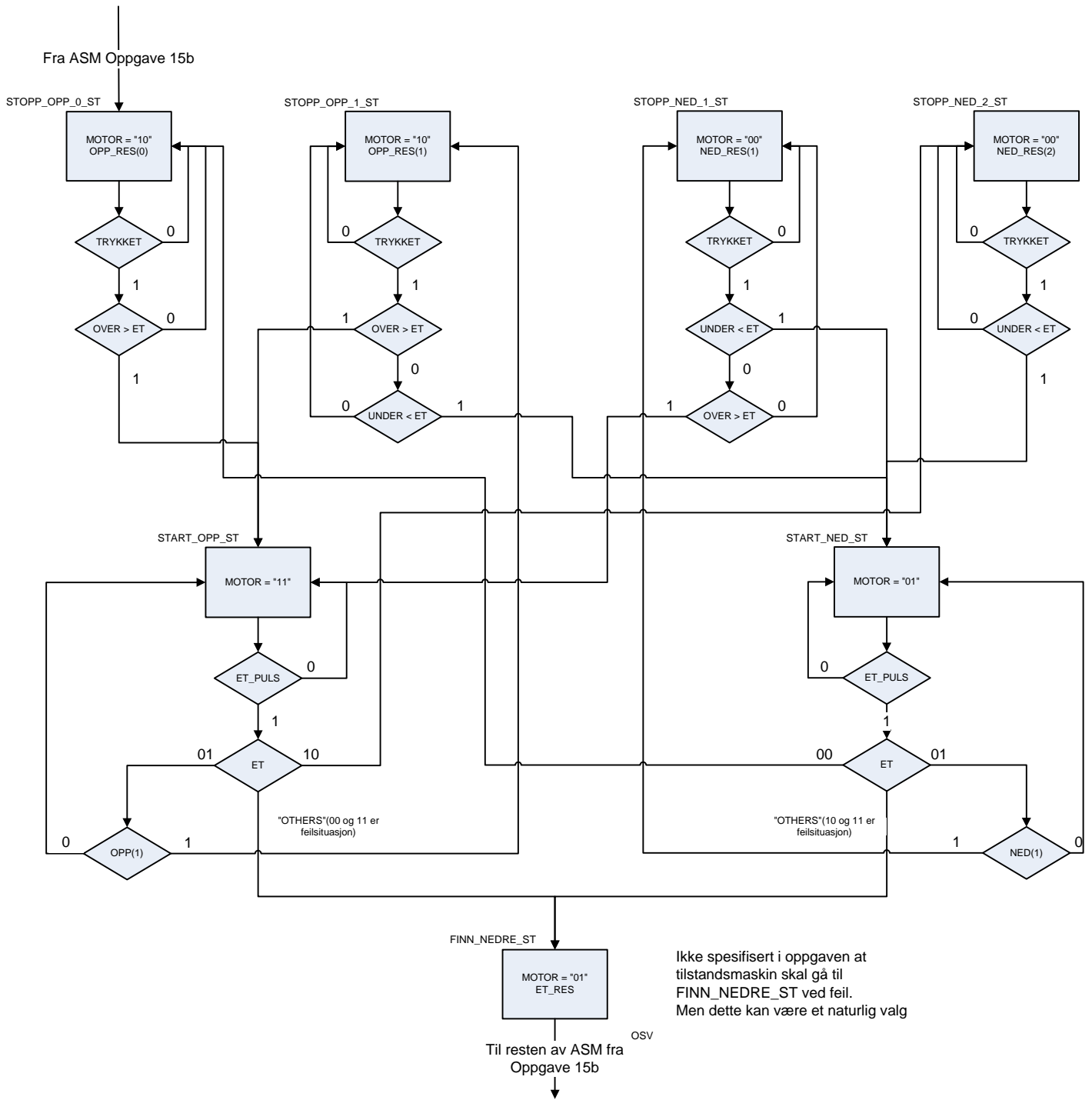
```
-- Start Oppgave 15a)
OVER_COMB:
process(opp,ned)
begin
  if unsigned(opp) /= 0 or unsigned(ned) /= 0 then
    trykket <= '1';
  else
    trykket <= '0';
  end if;

  over <= "00";
  if ned(2) = '1' then
    over <= "10";
  elsif opp(1) = '1' or ned(1) = '1' then
    over <= "01";
  elsif opp(0) = '1' then
    over <= "00";
  end if;
end process;
--End oppgave 15a)
```

b)



**c) Implementert som Moore maskin (Kan også implementeres som Mealy maskin)**



d)

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

architecture oppgave15_moore_rtl of oppgave15 is

    constant STOPP_NED : std_logic_vector(1 downto 0) := "00";
    constant START_NED : std_logic_vector(1 downto 0) := "01";
    constant STOPP_OPP : std_logic_vector(1 downto 0) := "10";
    constant START_OPP : std_logic_vector(1 downto 0) := "11";

    type heis_st_type is (FINN_NEDRE_ST,FINN_0_ST,STOPP_NEDRE_ST,
                        STOPP_OPP_0_ST);
    signal heis_st,heis_next_st : heis_st_type;

begin

    --Start Oppgave 15d)
    HEIS_ST_COMB:
    process(et,et_puls,nedre,heis_st)
    --Må legge til opp,ned,trykket på sensitivitetlisten
    --for komplett styring
    begin
        et_res <= '0';
        opp_res <= "00";
        ned_res <= "00";
        heis_next_st <= FINN_NEDRE_ST;

        case heis_st is
            when FINN_NEDRE_ST =>
                motor <= START_NED;
                opp_res <= "11";
                ned_res <= "11";
                heis_next_st <= FINN_NEDRE_ST;
                if nedre = '1' then
                    heis_next_st <= STOPP_NEDRE_ST;
                end if;

            when STOPP_NEDRE_ST =>
                motor <= STOPP_NED;
                et_res <= '1';
                heis_next_st <= FINN_0_ST;

            when FINN_0_ST =>
                motor <= START_OPP;
                et_res <= '1';
                heis_next_st <= FINN_0_ST;
                if et_puls = '1' then
                    heis_next_st <= STOPP_OPP_0_ST;
                end if;

            when STOPP_OPP_0_ST =>
                motor <= STOPP_OPP;
                opp_res(0) <= '1';
                heis_next_st <= STOPP_OPP_0_ST;
        end case;
    end process;

    -- Ikke nødvendig for oppgave 15b
    -- if trykket = '1' then
```

```

--      if over > et then
--          heis_next_st <= START_OPP_ST;
--      end if;
--  end if;

  when others =>
    motor <= STOPP_NED;
    heis_next_st <= FINN_NEDRE_ST;

  end case;

end process;

HEIS_ST_REG:
process(reset,clk)
begin
  if reset = '1' then
    heis_st <= FINN_NEDRE_ST;
  elsif rising_edge(CLK) then
    heis_st <= heis_next_st;
  end if;
end process;

end architecture oppgave15d_moore_rtl;

```

**e)**

Man kan benytte f.eks. signalet `et_puls` til å sette en teller til en verdi som kan lage en passende pause . Telleren teller ned til 0. Når telleren er 0 er signalet `timeout` aktivt. `timeout` benyttes som en betingelse sammen med trykket og `over/under` for å lage en forsinket start.