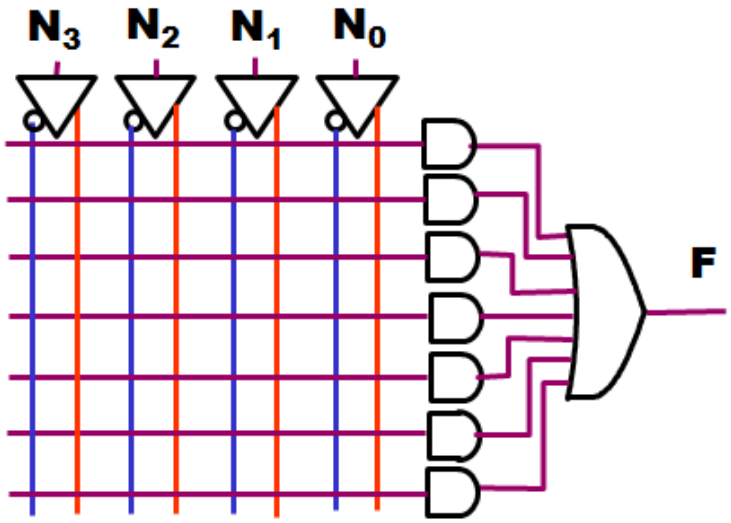
Eksempel – Primtals detektor

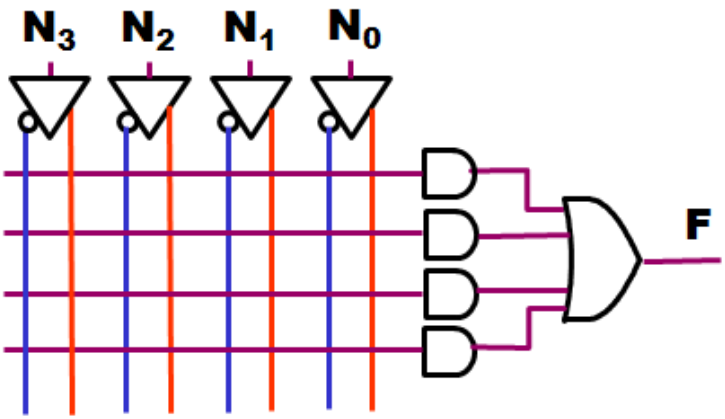
Lav et kredsløb der på grundlag af et 4-bit input bestemmer om N (et tal mellem 0 og 15) er et primtal
F=1 hvis N er et primtal ellers F=0



Logik til primtalsdetektor
uden reduktion
Se Wakerly figur 4-18 side 206

Sandhedstabel for Primtal-Detektor

	N_3	N_2	N_1	N_0	F
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	



Logik til primtalsdetektor
efter reduktion
Se Wakerly figur 4-25 side 212

N_1N_0 \ N_3N_2	00	01	11	10
0 0	0	4	12	8
0 1	1	5	13	9
1 1	3	7	15	11
1 0	2	6	14	10

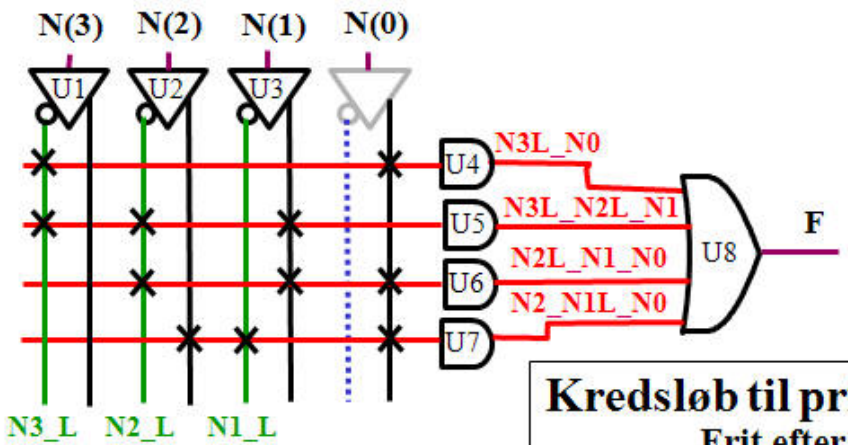
F=

```
architecture prime2_arch of prime is
  signal N3L_N0, N3L_N2L_N1, N2L_N1_N0, N2_N1L_N0: STD_LOGIC;
begin
   N3L_N0      <= not N(3)                                  and N(0);
   N3L_N2L_N1  <= not N(3) and not N(2) and      N(1);
   N2L_N1_N0   <=             not N(2) and      N(1) and N(0);
   N2_N1L_N0   <=                 N(2) and  not N(1) and N(0);

   F <= N3L_N0 or N3L_N2L_N1 or N2L_N1_N0 or N2_N1L_N0;
end prime2_arch;
```

Wakerly tabel 5-37 dataflow VHDL
Primtals detektor med boolske udtryk

```
architecture prime3_arch of prime is
  signal N3L_N0, N3L_N2L_N1, N2L_N1_N0, N2_N1L_N0: STD_LOGIC;
begin
   N3L_N0      <= '1' when N(3)='0' and N(0)='1' else
                  '0';
   N3L_N2L_N1  <= '1' when N(3)='0' and N(2)='0' and N(1)='1' else
                  '0';
   N2L_N1_N0   <= '1' when N(2)='0' and N(1)='1' and N(0)='1' else
                  '0';
   N2_N1L_N0   <= '1' when N(2)='1' and N(1)='0' and N(0)='1' else
                  '0';

   F <= N3L_N0 or N3L_N2L_N1 or N2L_N1_N0 or N2_N1L_N0;
end prime3_arch;
```

Wakerly tabel 5-38 dataflow VHDL
Primtals detektor med when else sætninger



Kredsløb til primtalsdetektor
Frit efter Wakerly

```
architecture prime4_arch of prime is
begin
    with N select
        F <= '1' when "0001",
             '1' when "0010",
             '1' when "0011"|"0101"|"0111"|"1011"|"1101",
             '0' when others;
end prime4_arch;
```

Wakerly tabel 5-40
with sætning med select signal angivelse

```
function CONV_INTEGER ( X: STD_LOGIC_VECTOR ) return INTEGER is
    variable RESULT: INTEGER;
begin
    RESULT := 0;
    for i in X'range loop
        RESULT := RESULT * 2;
        case X(i) is
            when '0'|'L' => null;
            when '1'|'H' => RESULT := RESULT + 1;
            when others => null;
        end case;
    end loop;
    return RESULT;
end CONV_INTEGER;
```

Wakerly tabel 5-25 function der konverterer en STD_LOGIC_VECTOR til en INTEGER

```
architecture prime5_arch of prime is
begin
    with CONV_INTEGER(N) select
        F <= '1' when 1|2|3|5|7|11|13,
             '0' when others;
end prime5_arch;
```

Wakerly tabel 5-41
with sætning med lettere læselig kode

```
architecture prime6_arch of prime is
begin
  process (N)
    variable N3L_N0, N3L_N2L_N1, N2L_N1_N0, N2_N1L_N0: STD_LOGIC;
  begin
    N3L_N0    := not N(3)                                  and N(0);
    N3L_N2L_N1 := not N(3) and not N(2) and      N(1);
    N2L_N1_N0 :=              not N(2) and      N(1) and N(0);
    N2_N1L_N0 :=                  N(2) and  not N(1) and N(0);
    F <= N3L_N0 or N3L_N2L_N1 or N2L_N1_N0 or N2_N1L_N0;
  end process;
end prime6_arch;
```

Wakerly tabel 5-43 Proces med boolske udtryk

```
architecture prime7_arch of prime is
begin
  process (N)
    variable NI: INTEGER;
  begin
    NI := CONV_INTEGER (N);
    if NI=1 or NI=2 then
      F <= '1';
    elsif NI=3 or NI=5 or NI=7 or NI=11 or NI=13 then
      F <= '1';
    else
      F <= '0';
    end if;
  end process;
end prime7_arch;
```

Wakerly tabel 5-45
Proces med **if** sætning

```
architecture prime8_arch of prime is
begin
  process (N);
  begin
    case CONV_INTEGER (N) is
      when 1 =>                  F <= '1';
      when 2 =>                  F <= '1';
      when 3 | 5 | 7 | 11 | 13 => F <= '1';
      when others               => F <= '0';
    end case
  end process;
end prime8_arch;
```

Wakerly tabel 5-47
Proces med **case** sætning

```
entity prime9 is
  port ( N: in   STD_LOGIC_VECTOR ( 7 downto 0);
         F: out  STD_LOGIC );

architecture prime9_arch of prime9 is
begin
  process (N)
     variable NI:     INTEGER;
     variable Prime:  BOOLEAN;
  begin
     NI    := CONV_INTEGER ( N );
     Prime := true;
     if NI=1 or NI=2 then
        null;
     else
        for i in 2 to 253 loop
           if (NI mod i = 0) and (NI /= i ) then
              prime := false;
              exit;
           end if;
        end loop;
     end if;
     if Prime then
        F <= '1';
     else
        F <= '0';
     end if;
  end process;
end prime9_arch;
```

Wakerly tabel 5-50
Primtals-detektor baseret på et "højniveau-program"

```
entity Prime8 is
    Generic( Max: Natural := 6);
    Port ( N : in  STD_LOGIC_VECTOR (Max downto 0);
           Prime : out  STD_LOGIC);
end Prime8;
```

```
36  architecture Behavioral of Prime8 is
37  begin
38  -- The process finds Primes by the the principles
39  -- of Eratostens Sive
40     Process(N)
41        type tabel is array (0 to 2**Max-1) of boolean;
42        variable Primtal,Tal: tabel;
43        variable Ni,j: integer range 0 to 2**Max;
44     begin
45        ------- Fill Tal with all numbers 0 to 2**Max
46        for i in Primtal'Range loop
47           Primtal(i) := False; -- Clear the Prime sive
48           Tal(i)     := True;
49        end loop;
50        -------- Remove 0,1 and set 2 as a Prime
51        Tal(0) := False;
52        Tal(1) := False;
53        Primtal(1) := True;
54        -- The Next (lowest) number will be a Prime
55        for i in 2 to 2**Max-1 loop
56           if Tal(i) then
57              Primtal(i) := True; -- Set as Prime
58              j := i;
59              while j<2**Max loop
60                 Tal(j) := False; -- Remove Multipla
61                 j := j +i;
62              end loop;
63           end if;
64        end loop;
65        ------------------------------------------------
66        Ni := conv_integer(N); -- Now decide if
67        if Primtal(Ni) then     -- N is a Prime number
68           Prime <= '1';
69        else
70           Prime <= '0';
71        end if;
72     end process;
73  end Behavioral;
```