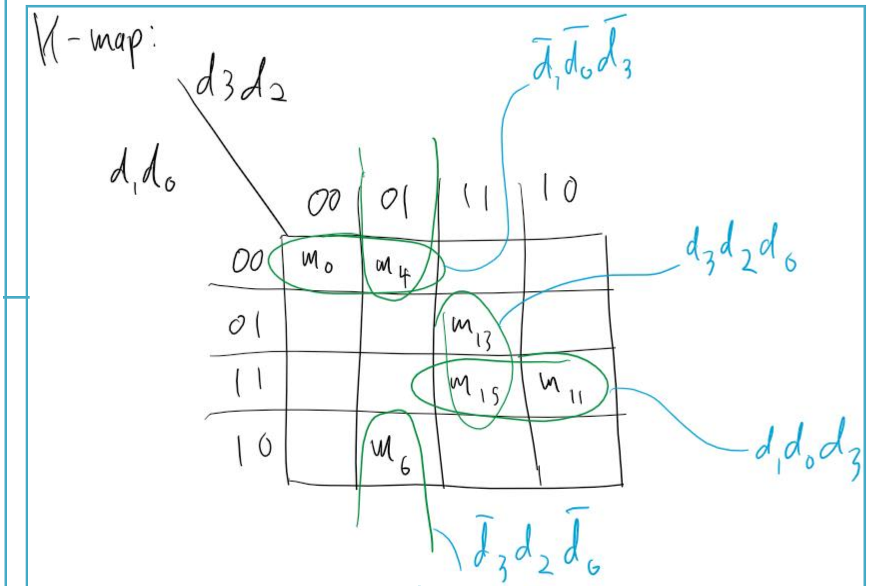
30139$\boxed{4066}$

$f(= \sum m(0, 4, 6, 11, 13, 15)$

d3	d2	d1	d0		
0	0	0	0	m0	—
0	0	0	1	m1	
0	0	1	0	m2	
0	0	1	1	m3	
0	1	0	0	m4	—
0	1	0	1	m5	
0	1	1	0	m6	—
0	1	1	1	m7	
1	0	0	0	m8	
1	0	0	1	m9	
1	0	1	0	m10	
1	0	1	1	m11	—
1	1	0	0	m12	
1	1	0	1	m13	—
1	1	1	0	m14	
1	1	1	1	m15	—

CSOP: $f = \overline{d0}\,\overline{d1}\,\overline{d2}\,\overline{d3} + \overline{d0}\,\overline{d1}\,d2\,\overline{d3} + \overline{d0}\,d1\,d2\,\overline{d3} + d0\,\overline{d1}\,d2\,d3 + d_0\,d_1\,\overline{d}_2\,d_3 + d0\,d1\,d2\,d3$

K-map:



Simplified Boolean Expression:

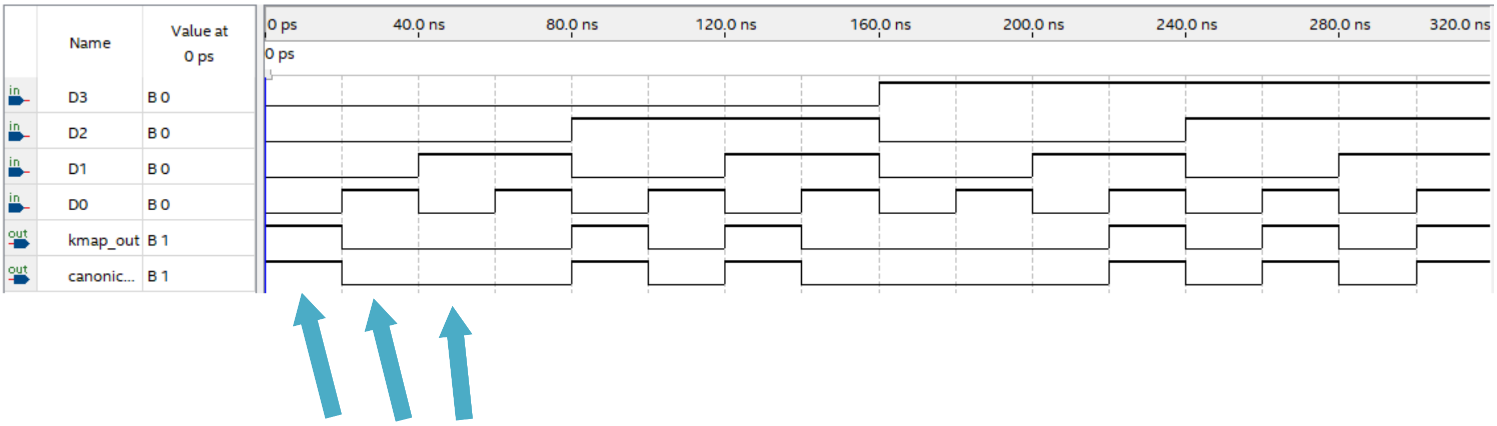
$$f = d_1d_0d_3 + d_3d_2d_0 + \bar{d}_1\bar{d}_0\bar{d}_3 + \bar{d}_3d_2\bar{d}_0$$

VHDL Code for circuit4066.vhd

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY circuit4066 IS
       PORT(D0, D1, D2, D3 : IN STD_LOGIC;
       canonical_out, kmap_out : OUT STD_LOGIC);
END circuit4066;

ARCHITECTURE Behaviour OF circuit4066 IS

BEGIN
       --unsimplified CSOP expression from truth table and minterms
       canonical_out <=      (not d0 and not d1 and not d2 and not d3) or --m0
                             (not d0 and not d1 and     d2 and not d3) or --m4
                             (not d0 and     d1 and     d2 and not d3) or --m6
                             (    d0 and not d1 and     d2 and     d3) or --m11
                             (    d0 and     d1 and not d2 and     d3) or --m13
                             (    d0 and     d1 and     d2 and     d3);  --m15

       --Simplified Boolean expression from the Karnaugh map
       kmap_out <= (d1 and d0 and d3) or
                             (d3 and d2 and d0) or
                             (not d1 and not d0 and not d3) or
                             (not d3 and d2 and not d0);

END Behaviour;
```

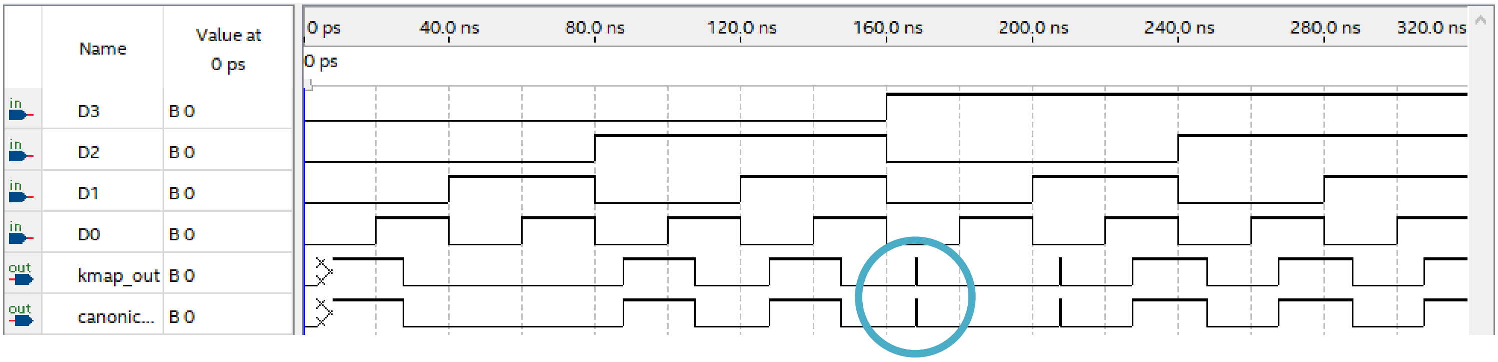
Waveform Simulation Results

Functional Simulation



minterms 0, 1, 2, and so on

Timing Simulation



Discussion

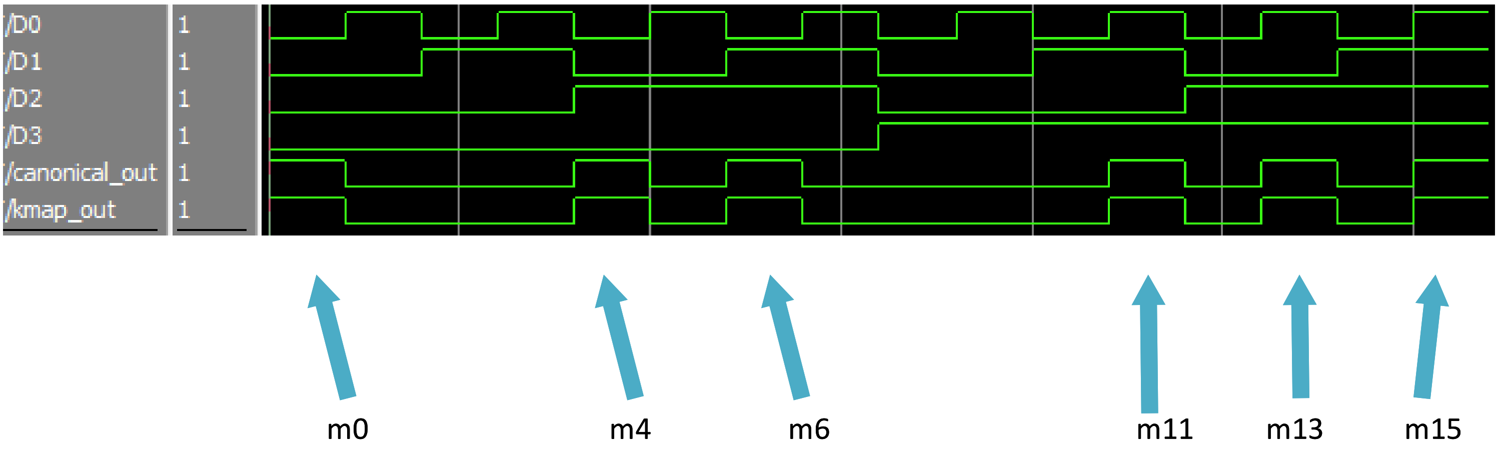
In both simulations, D0-D3 alternate in such intervals as to cover every row of a four-input truth table, with D0 alternating at the highest frequency and D3 only changing value once, halfway through the simulation. The minterms are represented in order from 0 to 15, and are divided by the dashed vertical lines. In the functional simulation, kmap_out and canonical_out both exhibit the same (desired) behaviour; namely, they switch to 1 for minterms 0, 4, 6, 11, 13, and 15. The timing simulation has correct results as well, but due to the inherent latency of the simulated hardware, it takes about 8 ns for the changes to propagate to the outputs. Other anomalies include the brief spikes around 170 and 210 ns, probably indicating that the circuit has some hysteresis, as the spike would make sense if the inputs which switched to 0 had remained at 1. For example, at 170 ns, if D0, D1, and D2 had remained at 1, when D3 switched to 1 the circuit is effectively simulating minterm 15, and thus should produce an output of 1.

VHDL Code for circuit4066.vhd

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

--testbenches have empty entity sections
ENTITY tb_circuit4066 IS
END tb_circuit4066;


ARCHITECTURE test of tb_circuit4066 is

COMPONENT circuit4066 IS
PORT(D0, D1, D2, D3 : IN STD_LOGIC; --inputs
    canonical_out, kmap_out : OUT STD_LOGIC); --outputs
END COMPONENT;

--assigning signals for each port to stimulate them
SIGNAL sig0, sig1, sig2, sig3 : STD_LOGIC;
SIGNAL canonsig, ksig : STD_LOGIC;


--instantiate the DUT
BEGIN
DUT : circuit4066
PORT MAP(D0 => sig0, D1 => sig1, D2 => sig2, D3 => sig3,
         canonical_out => canonsig, kmap_out => ksig);


--code to systematically iterate through all possible values
--of the input variables:
PROCESS IS
BEGIN

--sig0 alternates every 20 ns
sig0 <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns, '0' after 80 ns,
        '1' after 100 ns, '0' after 120 ns, '1' after 140 ns, '0' after 160 ns,
        '1' after 180 ns, '0' after 200 ns, '1' after 220 ns, '0' after 240 ns,
        '1' after 260 ns, '0' after 280 ns, '1' after 300 ns;

--sig1 alternates every 40 ns
sig1 <= '0', '1' after 40 ns, '0' after 80 ns, '1' after 120 ns, '0' after 160 ns,
        '1' after 200 ns, '0' after 240 ns, '1' after 280 ns;

--sig2 alternates every 80 ns
sig2 <= '0', '1' after 80 ns, '0' after 160 ns, '1' after 240 ns;

--and sig3 only alternates every 160 ns
sig3 <= '0', '1' after 160 ns;

WAIT;

END PROCESS;
END test;
```

ModelSim Waveform Simulation Results

Simulated waveforms



Discussion

The waveforms here are identical to those shown in the functional simulation from the original VHDL code – which is a good sign that my testbench code is correct! The waveforms are listed here from LSB to MSB, but beyond that they are identical, and the two outputs produce 1's at the desired minterms (0, 4, 6, 11, 13, 15).