A_1

Assignment 1

CS 3482; Professor Tang

Connor Taffe. T no. 3742

April 21st, 2015

1 Question

Draw the circuit in LogicWorks to implement the Boolean function:

$$f(a, b, c, d) = (a' + b)'c + d(b' + ac)$$

Figure 1: Given boolean function f

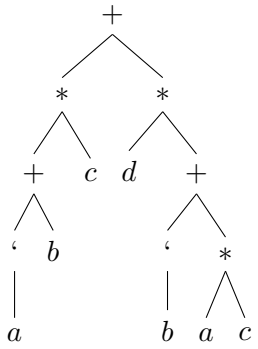

Figure 2: Parse tree of figure 1

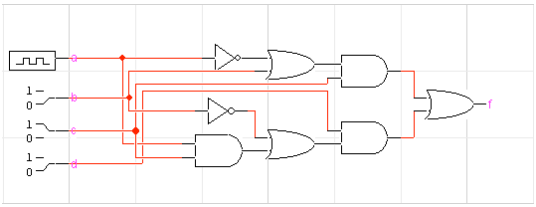

Figure 3: LogicWorks implementation of f

(a) Figure 2 shows the structure of a naive circuit implementation. Figure 3 shows the LogicWorks implementation of the circuit.

(b) I observed an 18ns delay between a and f (shown in figure 4). This is because the gate delays between a and f accumulate.

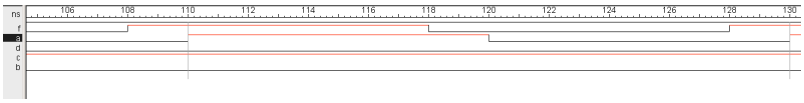

Figure 4: LogicWorks implementation of f, waveforms

2 Question

Simplify the same Boolean function in figure 1 to be the sum of its minterms by obtaining is truth table first.

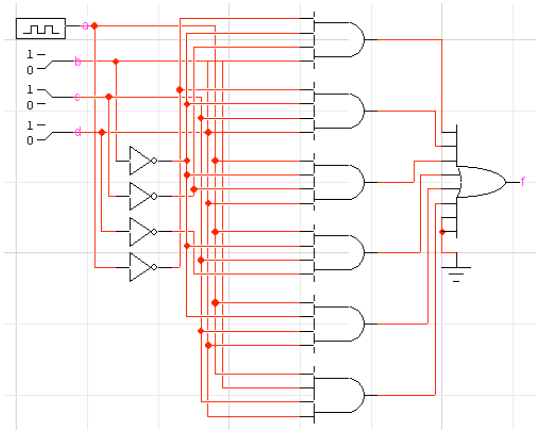

Figure 5: LogicWorks implementation of f_{minterm}

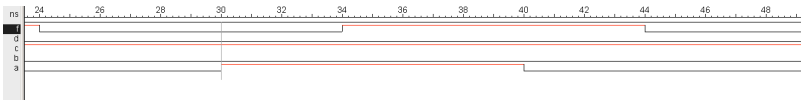

Figure 6: LogicWorks implementation of f_{minterm}, waveforms

(a) Figure 1 shows the truth table for the function.

(b) $f_{\text{minterm}} = a'b'c'd + a'b'cd + ab'c'd + ab'cd' + ab'cd + abcd$

(c) Figure 7 shows the parse tree and thusly the gate implementation of the minterms of f. The LogicWorks circuit is represented in figure 5.

(d) I observed a delay of 4ns between a and f (shown in figure 6). This is because the delays between a and f accumulate as in the naive implementation, but less so as there is a maximum of three gates in the path beween a and f.

3 Question

(a) Figure 8 shows the trees of a 2-to-4 decoder with an active-high enable E. Testing is shown in figure 10, and the tested circuit LogicWorks implementation is shown in figure 9.

a	b	c	d	a'	$a'+b$	$(a'+b)'$	$(a'+b)'c$	ac	b'	$b'+ac$	$d(b'+ac)$	f
0	0	0	0	1	1	0	0	0	1	1	0	0
0	0	0	1	1	1	0	0	0	1	1	1	1
0	0	1	0	1	1	0	0	0	1	1	0	0
0	0	1	1	1	1	0	0	0	1	1	1	1
0	1	0	0	1	1	0	0	0	0	0	0	0
0	1	0	1	1	1	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	1	0	0	1	1	1	1
1	0	1	0	0	0	1	1	1	1	1	0	1
1	0	1	1	0	0	1	1	1	1	1	1	1
1	1	0	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	1	0	1	1	1

Table 1: Truth table for f as described in figure 1

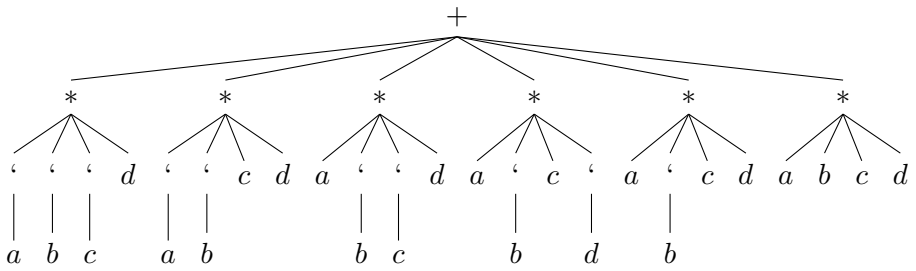

Figure 7: Parse tree of minterms of f

(b) Testing is shown in figure 12, and the tested circuit LogicWorks packaged implementation is shown in figure 11.

4 Question

(a) Five, one for the first two bits, and four for each of the four minterms of the first two bits.

(b) Figure 15 shows the parse trees of a 4-to-16 decoder built only using 2-to-4 decoders. The parse tree describes the gate diagram. The LogicWorks implementation can be seen in figure 13.

(c) Figure 14 shows the waveform.

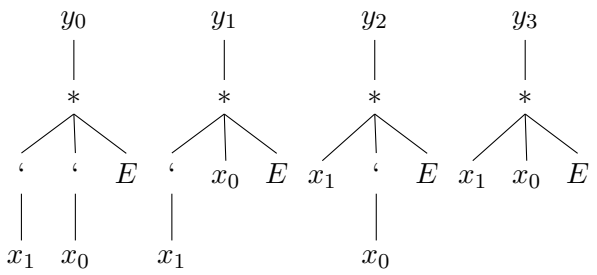

Figure 8: Parse trees of 2-to-4 decoder with an active-high enable E

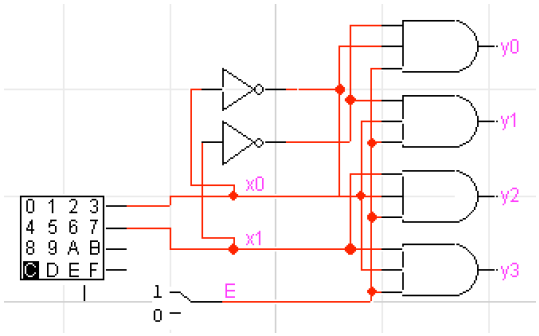

Figure 9: Internal circuit of a 2-4 decoder; being tested.

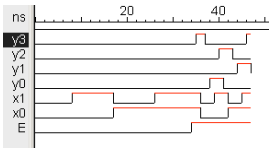

Figure 10: Test waveform output of a 2-4 decoder.

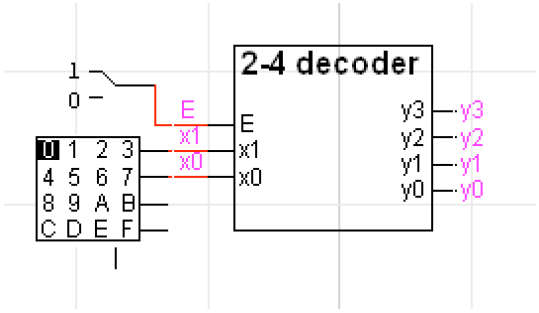

Figure 11: Internal circuit of a 2-4 decoder (packaged); being tested.

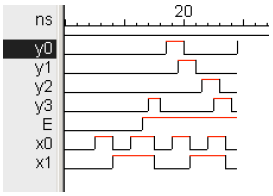

Figure 12: Test waveform output of a 2-4 decoder (packaged).

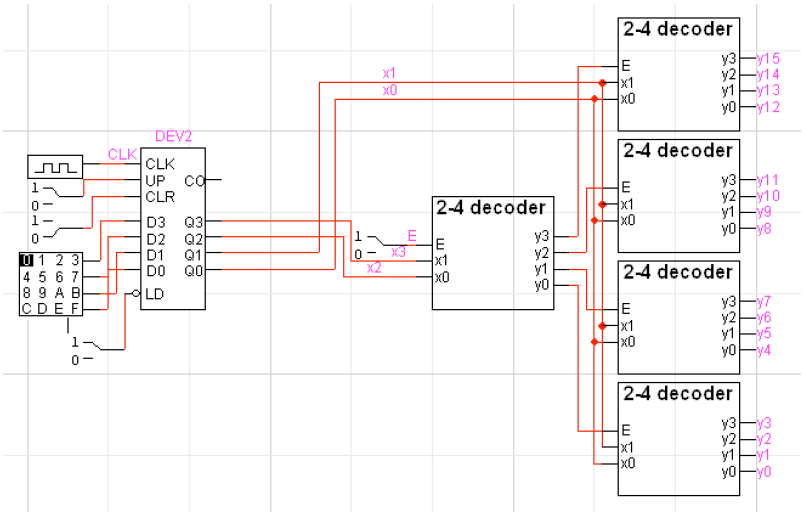

Figure 13: Internal circuit of a 4-16 decoder; being tested.

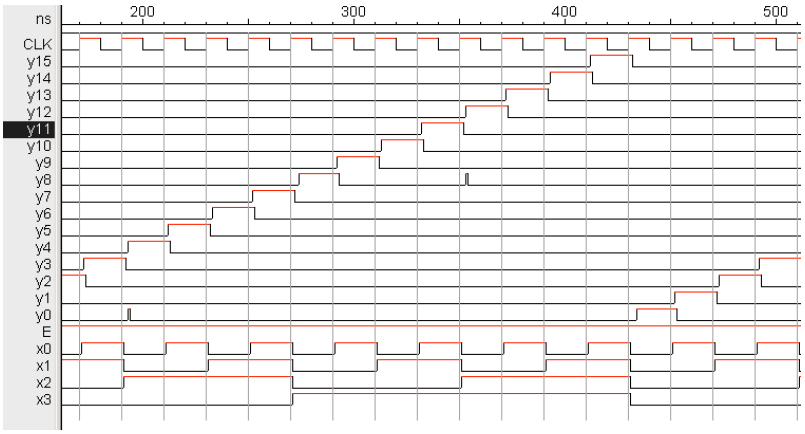

Figure 14: Test waveform output of a 4-16 decoder.

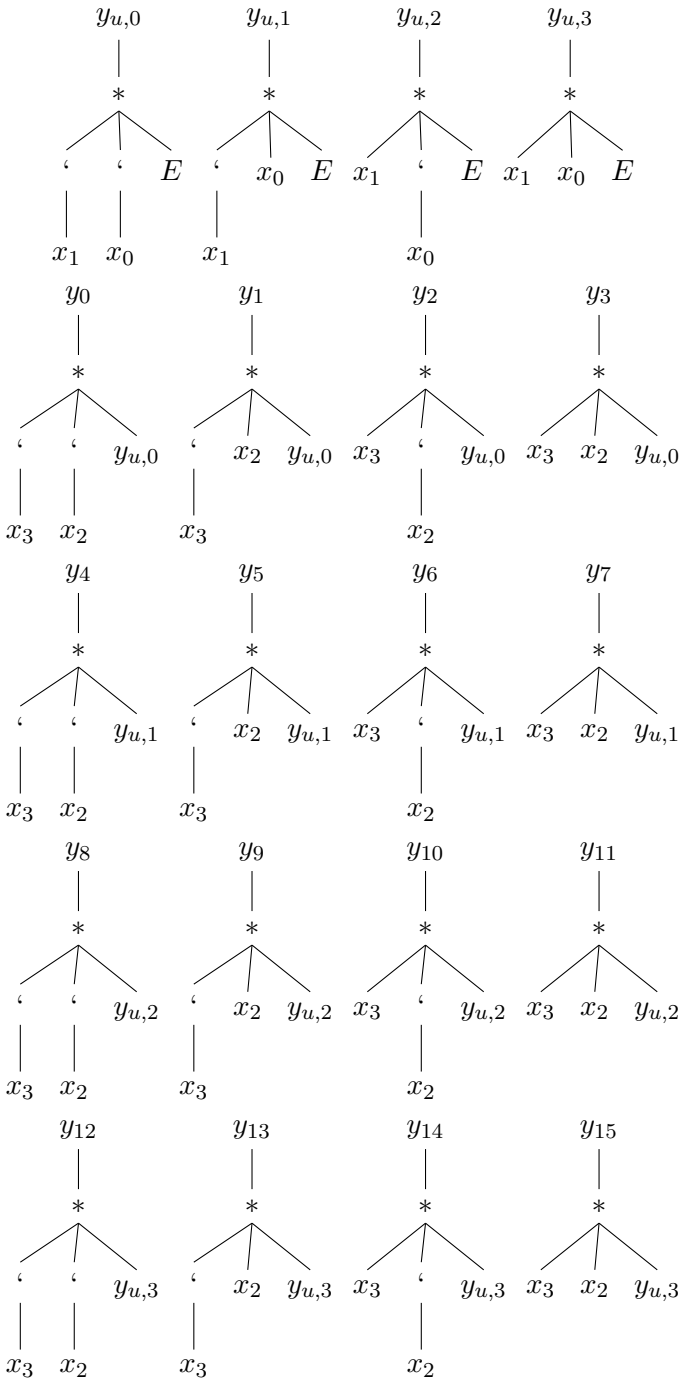

Figure 15: Parse trees of 4-to-16 decoder with an active-high enable E. For convenience and space the first line depicts the inner 2-to-4 decoder for the first two bits, denoted with subscript u.