

SYSC2310 Lab 4 Report

1.0 EXERCISE 1: DESIGN AN ARBITRARY FUNCTION

1.1 Description of Circuit's Operation

The circuit designed in Exercise 1 has four inputs and two outputs. The input is a 4-bit number. One of the outputs signals HIGH when the inputted number is prime and outputs LOW when it is not. The other output signals HIGH when the inputted number is divisible by 3 or 4 and signals LOW when the number meets neither of those conditions.

The circuit was designed using Logisim's Combinational Analysis function. The truth table for the circuit was inputted manually, the kmaps were generated, and the circuit diagram was produced by the program, shown in Figure 1.

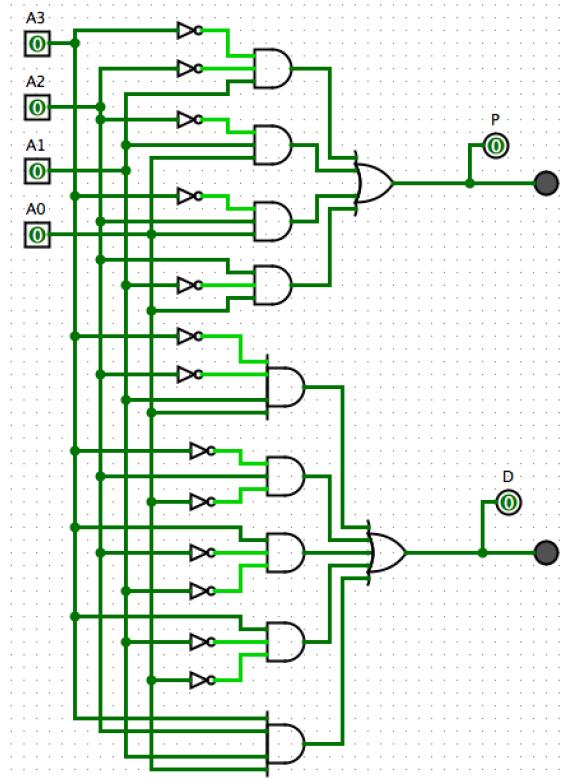


Figure 1. Diagram of Exercise 1 circuit

1.2 Verifying the Circuit's Operation

The circuit's operation was verified using Logisim's logging function, shown in Figure 2, by comparing actual outputs with their expected outputs.

A3	A2	A1	A0	LED(420,140)	LED(420,430)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	1	0	0	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	1	1	0
1	1	1	1	0	1
1	1	0	1	1	0
1	1	0	0	0	1
1	0	0	0	0	1
0	0	0	0	0	0
0	0	1	0	0	0
0	1	1	0	0	1
0	1	1	1	0	1
0	1	1	1	1	0
1	1	1	1	0	1
1	0	1	1	1	0
0	0	1	1	1	0

Figure 2. Logging file used to verify circuit operation in Exercise 1

2.0 EXERCISE 2: BCD TO 7-SEGMENT DECODER

2.1 Description of Circuit's Operation

The circuit designed in Exercise 2 is a BCD to 7-segment decoder. As shown in Figure 3, the circuit takes in 4 inputs, representing a binary number and has 7 outputs which go to a 7-segment display, displaying the corresponding input between 0-9 on the 7-segment display. The circuit works in a desired manner only for inputs between 0-9. It has been observed that inputting any number between 10-15 will output either a 9, 5, mirrored 6, or 6, on the display, however the inputs are being falsely represented.

The circuit was designed using Logisim's Combinational Analysis function. The truth

table for the circuit was inputted manually, the K-maps were generated, and the circuit diagram was produced by the program.

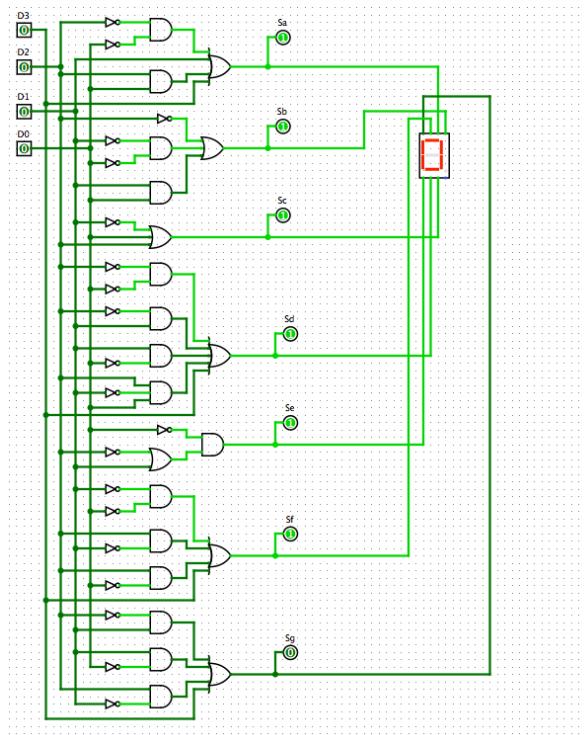


Figure 3. Diagram for the circuit in Exercise 2

2.2 Verifying the Circuit's Operation

The circuit's operation was verified using Logisim's logging function, shown in Figure 4, by comparing actual outputs with their expected outputs.

D3	D2	D1	D0	Sa	Sb	Sc	Sd	Se	Sf	Sg
1	1	1	1	1	1	1	1	0	1	1
1	1	0	1	1	0	1	1	0	1	1
1	1	0	0	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	0
0	1	0	0	0	1	1	0	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
0	1	0	1	1	0	1	1	0	1	1
1	1	0	1	1	0	1	1	0	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1

Figure 4. Logging file used to verify circuit operation in Exercise 2

INTRODUCTION TO DIGITAL SYSTEMS

SYSC2310 (Fall 2020)

Dr. Mostafa Taha

Lab 3: Answer Sheet

Exercise 1: Design an Arbitrary Function.

$$P = \overline{A_2} A_1 A_0 + \overline{A_3} \overline{A_2} A_1 + \overline{A_3} A_2 A_0 + A_2 \overline{A_1} A_0$$

$$D = \overline{A_3} A_2 A_0 + A_3 \overline{A_2} \overline{A_1} + A_3 \overline{A_1} \overline{A_0} + \overline{A_3} \overline{A_2} A_1 A_0 + A_3 A_2 A_1 A_0$$

The total number of AND and OR gates is:
3-pin AND: 4-pin AND: 4-pin OR: 5-pin OR:
7 2 1 1

Exercise 2: BCD to 7-Segment Decoder (IC 7447)

$$S_a = \overline{D_2} \overline{D_0} + D_3 + D_1 + D_2 D_0$$

$$S_b = \overline{D_2} + \overline{D_1} \overline{D_0} + D_1 D_0$$

$$S_c = \overline{D_1} + D_0 + D_2$$

$$S_d = \overline{D_2} D_0 + \overline{D_2} D_1 + D_1 \overline{D_0} + D_3 + D_2 \overline{D_1} D_0$$

$$S_e = \overline{D_0} (\overline{D_2} + D_1)$$

$$S_f = \overline{D_1} \overline{D_0} + D_2 \overline{D_0} + D_2 \overline{D_1} + D_3$$

$$S_g = D_2 \overline{D_1} + D_3 + \overline{D_2} D_1 + D_1 \overline{D_0}$$

The total number of AND and OR gates is:

2-pin AND: 3-pin AND: 4-pin OR: 3-pin OR: 5-pin OR: 2-pin OR:
14 1 3 2 1 1

Part 1A

SPSC2221D Lab 3 - Exercise 1 : PART 1A

= 4 inputs, 2 outputs

- let A3, A2, A1, AD be w, x, y, z respectively

A3	A2	A1	AD	P	D
w	x	y	z	0	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	1
1	1	1	1	1	1

r { true number is PRIME
false, if not PRIME

D { true if divisible by 3 or 4
false, "

Find P

wx	yz	00	01	11	10
00	-	1	1	-	-
01	-	-	1	1	-
11	-	-	-	1	-
10	-	-	-	-	1

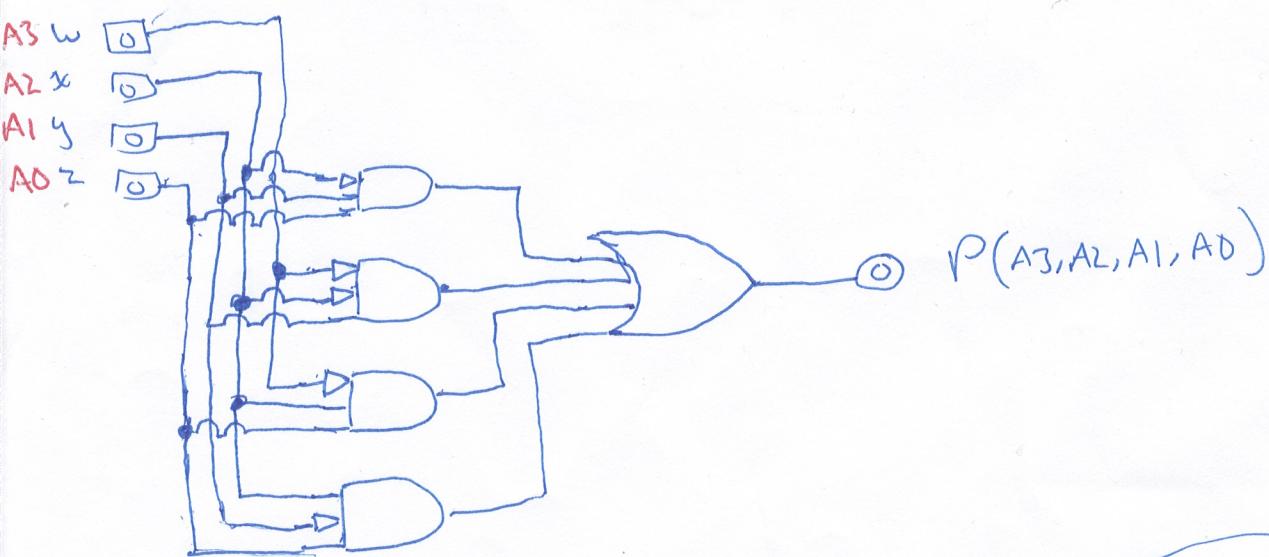
$$P(w, x, y, z) = x'y'z + w'x'y + w'x'z + xy'z$$

Find D

wx	yz	00	01	11	10
00	-	1	1	1	1
01	-	1	1	1	1
11	-	1	1	1	1
10	-	1	1	1	1

$$D(w, x, y, z) = w'x'z + wx'y' + wy'z' + w'x'y'z + wxy'z$$

Find P again

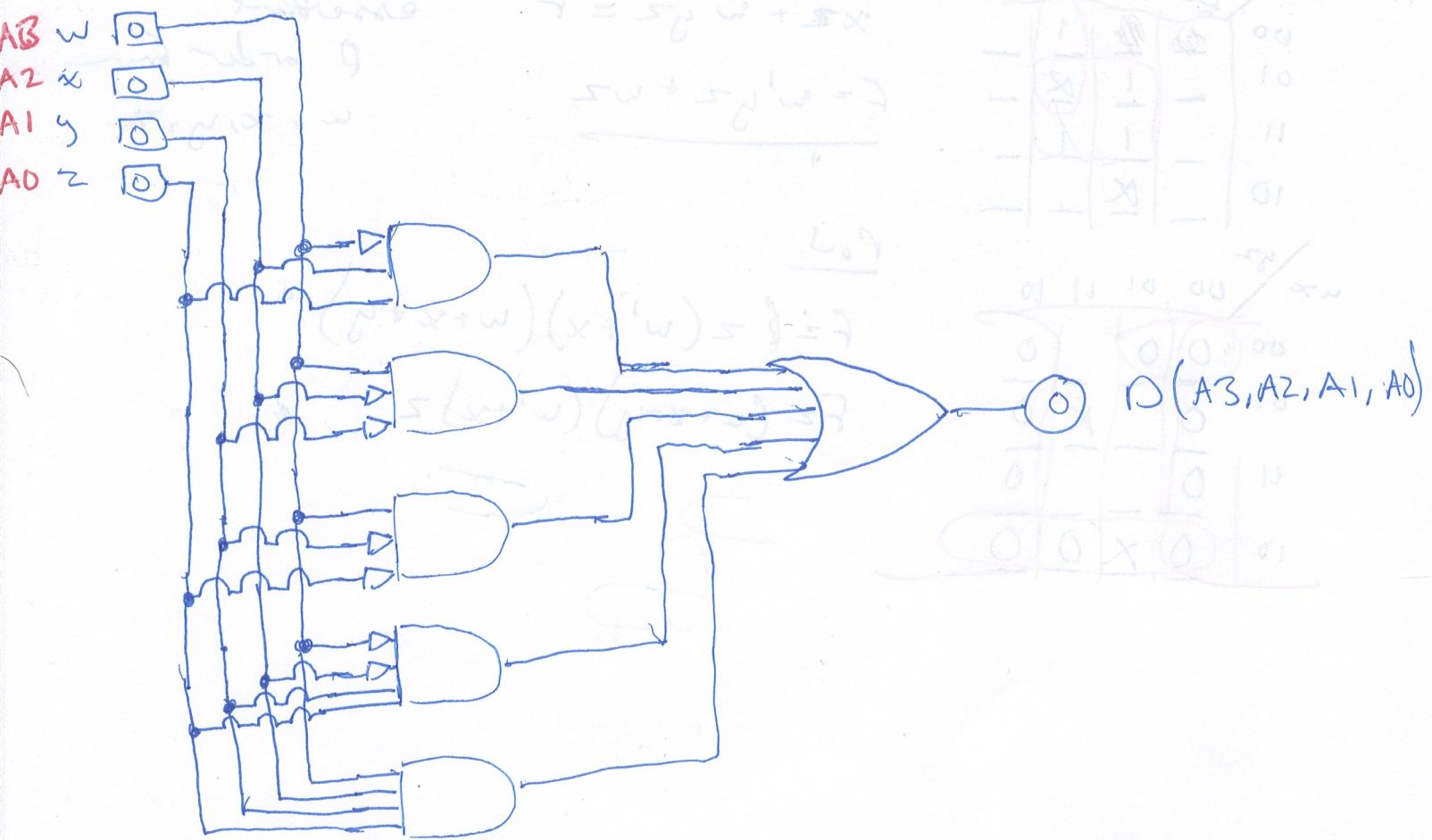


$$P(A3, A2, A1, AD)$$

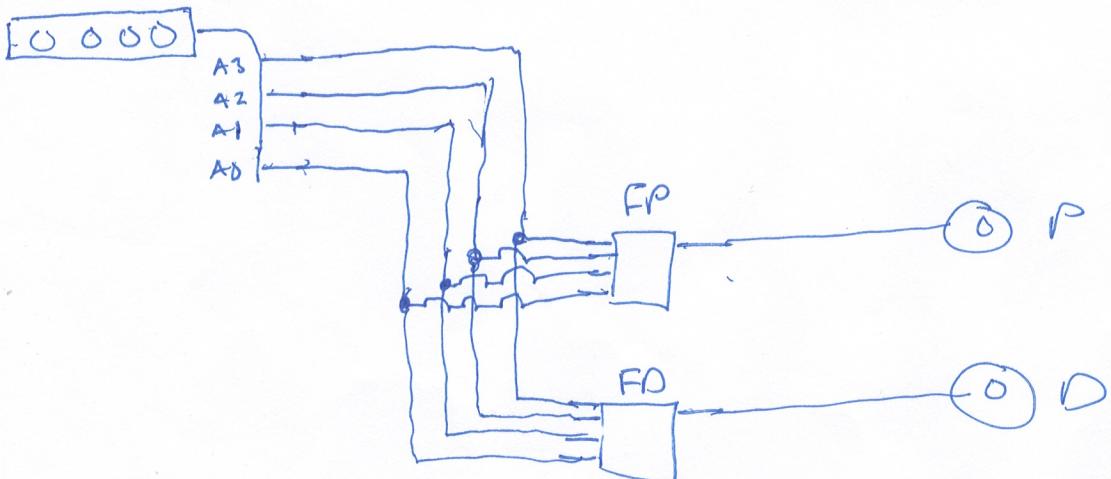
PART 1A

page 1

Function D diagram



Main Program



PART IA page 2

Counting the # of gates

3-input AND: 7

4-input AND: 2

4-input OR: 1

5-input OR: 1

Input		Output	
0	0	0	0
0	0	0	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

white school out
Liberty 40V

$$S' \oplus A + J \oplus A + S' \oplus A + S' \oplus A + (S \oplus A)_{\text{not}}$$

↑ shows 2 bits not, instead of subtract
to 8 bits & count, goes to 32 bits for bank add

$$J \oplus K \oplus A =$$

improved term

1st term

2nd term

3rd term



last one

PART 1A

page 3

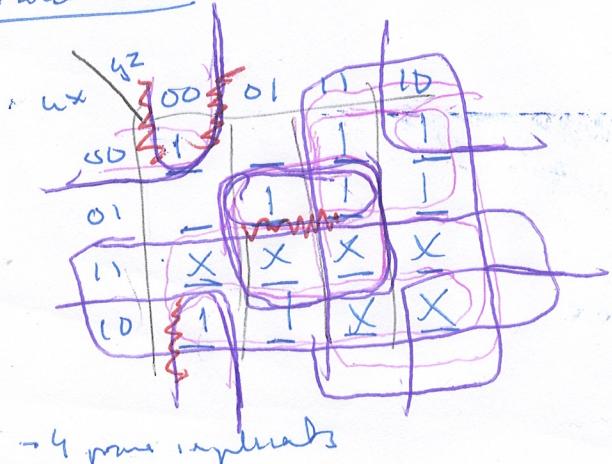
PART 2A

SYSC237D - LAB3 - Exercise 2: PART 2A

- let w, x, y, z be D3, D2, D1, D0

D3	D2	D1	D0	w	x	y	z	S _a	S _b	S _c	S _d	S _e	S _f	S _g
0	0	0	0	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1	1	0	1	0
0	0	1	1	1	1	1	1	0	0	0	1	0	0	1
0	1	0	0	0	1	1	0	0	0	1	1	1	1	1
0	1	0	1	1	0	1	1	0	1	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X

Find the S_a

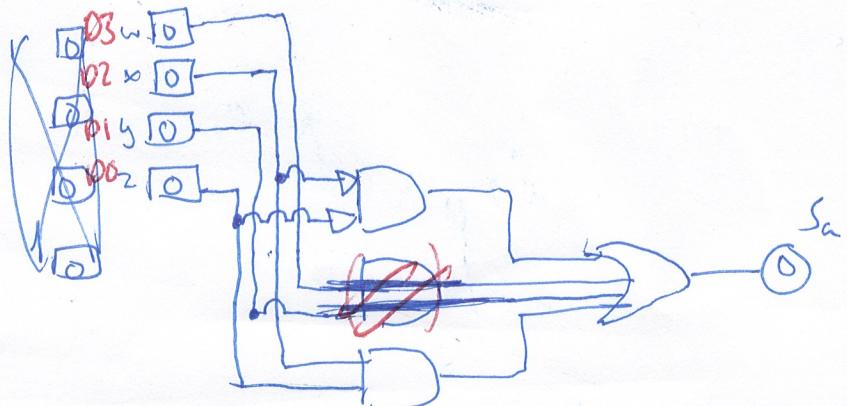


→ 4 prime implicants

Now S_a:

$$S_a(w, x, y, z) = x'z' + w + y + xz$$

$$S_a(D_3, D_2, D_1, D_0) = D_2'D_0D_1 + D_3 + D_1 + D_2D_0$$



Function S_b

wx	yz	00	01	11	10
00	1	1	1	1	1
01	1	0	1	0	1
11	X	X	X	X	X
10	1	1	X	X	X

-> prime implicants

using SOP:

$$S_b(w, x, y, z) = w' + y'z' + yz$$

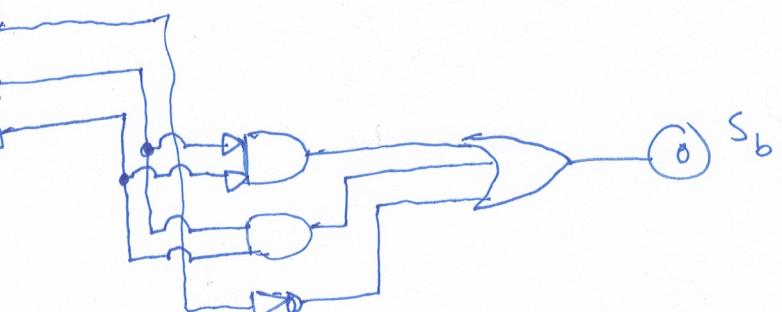
$$S_b(D_3, D_2, D_1, D_0) = D_2' + D_1'D_0' + D_1D_0$$

D₃ w

D₂ x

D₁ y

D₀ z



Function S_c

wx	yz	00	01	11	10
00	1	1	1	1	1
01	1	1	1	1	1
11	X	X	X	X	X
10	1	1	X	X	X

-> prime implicants

using SOP:

$$\Rightarrow S_c(w, x, y, z) = y' + z + x$$

$$S_c(D_3, D_2, D_1, D_0) = D_1' + D_0 + D_2$$

D₃ w

D₂ x

D₁ y

D₀ z



Function S_d

wx	yz	00	01	11	10
00	1	1	1	1	1
01	1	1	1	1	1
11	X	X	X	X	X
10	1	1	X	X	X

-> prime implicants

$$S_d(w, x, y, z) = x'z + x'y + yz' + w + xy'z$$

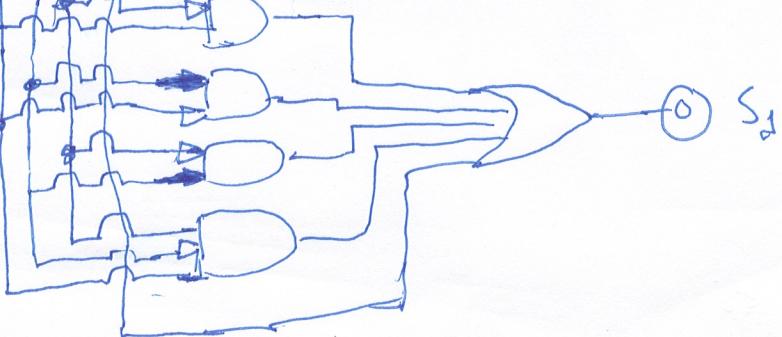
$$S_d(D_3, D_2, D_1, D_0) = D_2'D_0 + D_2'D_1' + D_1D_0' + D_3 + D_2D_1'D_0$$

D₃ w

D₂ x

D₁ y

D₀ z



Function S_e

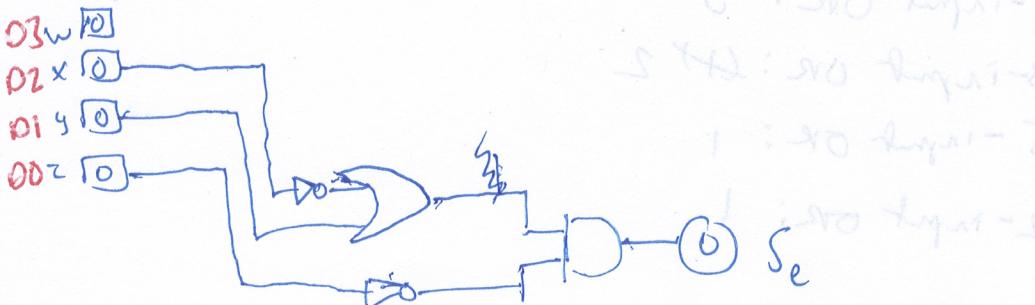
wx	yz	00	01	11	10
00	1	0	0	1	1
01	0	0	0	1	1
11	X	X	X	X	X
10	1	0	X	X	X

- 2 inputs

Using $P_0, S:$

$$S_e(w, x, y, z) = z'(x' + y)$$

$$S_e(D_3, D_2, D_1, D_0) = D_0'(D_2' + D_1)$$



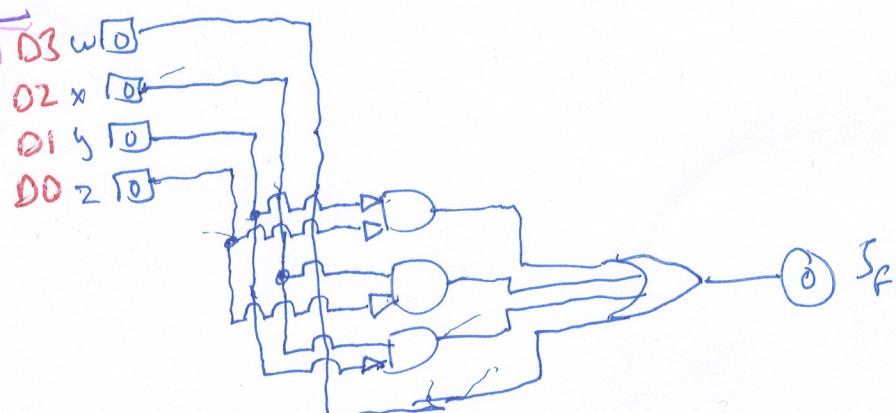
Function S_f

wx	yz	00	01	11	10
00	1	1	1	1	1
01	1	1	1	1	1
11	X	X	X	X	X
10	1	1	X	X	X

- 4 inputs

$$S_f(w, x, y, z) = y'z' + xz' + xy' + w$$

$$S_f(D_3, D_2, D_1, D_0) = D_1'D_0' + D_2D_0' + D_2D_1' + D_3$$



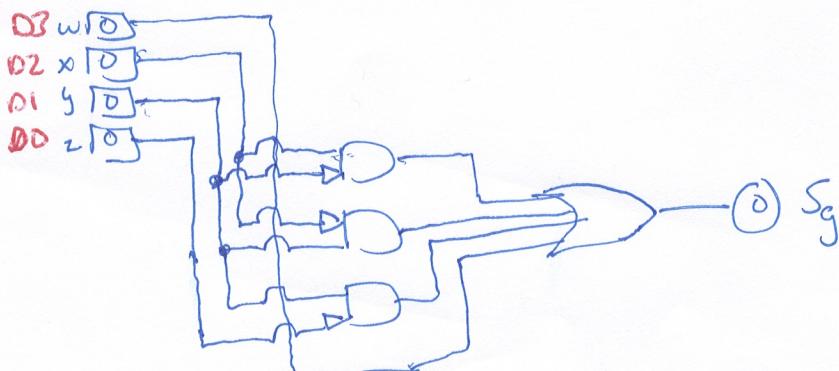
Function S_g

wx	yz	00	01	11	10
00	1	1	1	1	1
01	1	1	1	1	1
11	X	X	X	X	X
10	1	1	X	X	X

- 4 inputs

$$S_g(w, x, y, z) = xy' + w + x'y + yz'$$

$$S_g(D_3, D_2, D_1, D_0) = D_2D_1' + D_3 + D_2'D_1 + D_1D_0'$$



Counting # of gates:

2-input AND: 14

3-input AND: 1

4-input OR: 3

3-input OR: 6+2

5-input OR: 1

2-input OR: 1



2.9 pm

3b worked

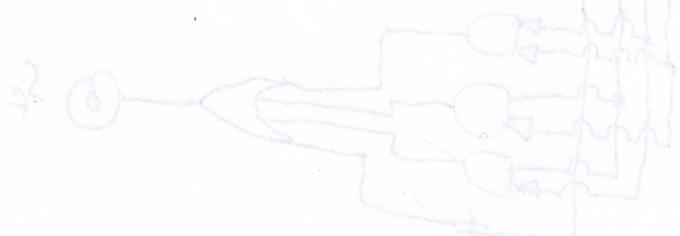
x_1	x_2	x_3	x_4	x_5	x_6
1	0	0	1	0	0
1	0	0	0	1	0
1	x	x	x	x	1
0	x	x	0	1	0
0	x	1	0	0	0
0	1	0	0	0	0

Strategy

2 min

$$w + \{px + \bar{s}x + \bar{s}'p\} = (\bar{w}psw)_{\bar{2}}$$

$$\bar{s}a + \{10sa + \bar{s}as a + \bar{s}a'1a\} = (\bar{a}as1as0, sa)_{\bar{2}}$$



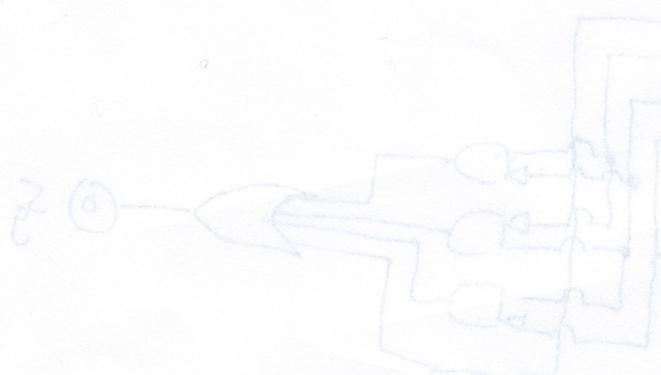
x_1	x_2	x_3	x_4	x_5	x_6
1	0	0	1	0	0
1	0	0	0	1	0
1	x	x	x	x	1
0	x	x	0	1	0
0	x	1	0	0	0
0	1	0	0	0	0

Strategy

2 min

$$w + \{px + \bar{s}x + w + \bar{p}x\} = (\bar{w}psw)_{\bar{2}}$$

$$\bar{s}a + \{10sa + \bar{s}as a + \bar{s}a'1a\} = (\bar{a}as1as0, sa)_{\bar{2}}$$



x_1	x_2	x_3	x_4	x_5	x_6
1	0	0	1	0	0
1	0	0	0	1	0
1	x	x	x	x	1
0	x	x	0	1	0
0	x	1	0	0	0
0	1	0	0	0	0

Strategy

2 min