FESC Laboratory: Bread Board TTL Full Adder

The aim of this lab is to introduce prototype boards, practice constructing circuits and to develop better debugging skills. As such there are no answers.

Task 1

Q: Try measuring the logic gates output voltage using a multi-meter with the LED unconnected. Does the output voltage increase?

A: yes, all logic gates have a limited fan-out (output current), increasing load will reduce output voltage.

Q: what will happen if a 330Ω or 600Ω resistor is used instead of a 470Ω i.e. how will it change the LED brightness? Try replacing 470Ω resistor with a higher value. Next try connecting the output LED directly to the output of the AND gate i.e. with no series resistor. Why isn't it damaged?

Question: why will connecting the LED with a 50Ω series resistor across the +5V / 0V supply rails be very bad?

A: key aim here is to show how an LED should be connected into a circuit without damaging the LED or the logic used to drive it. Reducing the series resistor will increase the current flow in the LED, increasing its brightness. Eventually too much current will flow through the LED, damaging the Silicon used in the diode. Assuming 2V across the LED using a 47Ω resistor will result in 64mA flowing through the LED (normally 10mA), which is bad. Connecting an LED directly to the logic gate's output isn't good, but, its not too bad as the internal 130Ω resistor will limit the current, but will put a lot of stress on the output driver i.e. it may get hot, shorten its life. Connecting the LED across the supply rails means there is no current limiting, so it will go POP.

Q: what is the output resistance of the logic gate when driving a logic '1' or a logic '0' onto a wire connected to its output? Why will these different resistances cause the time for the output to reach a stable logic state to vary?

A: logic '1' has to pass through the collector resistor 130Ω, a transistor and a diode. These will all limit the output current i.e. flow of electrons. When driving the output to a logic '0' there is only a single transistor to ground so there is less restrictions to the flow of current. The speed at which a signal can be driven onto a wire is determined by the rate of change of electrons, therefore, for this logic gate switching to a logic '1' takes longer than a logic '0'.

Q: to simplify analysis the 0.7V voltage drops caused by the output diode (D4) and transistor (Q5) are ignored the output resistance of the logic gates is equivalent to a 130Ω resistor tied to +5V. If each logic gate has an input resistance of 5KΩ, and the logical '1' threshold is 2V, how many inputs can be driven by one output?

A: using PD theorem to produce a 2V output with 130Ω top resistor you need a 87Ω bottom resistor, therefore to produce this resistance you need 57 5K resistors in parallel i.e. fan-out is 57-ish.

Task 2

Y = AB

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level
L = Low Logic Level

7408 : AND Gate

Q: construct the circuit shown in figure 5, remove the pull down resistors R1 and R2. Does the output change? What does that indicate about the state of the inputs?

A: to drive a logic '0' onto an input you actually have to suck current out of the input. Removing the pull-down resistors stops this, therefore, the inputs should float high i.e. take on a logic '1' state, setting the output high.

Q: replace the pull down resistors R1 and R2 with 47KΩ. Does the output change i.e. return to a logic '0'? What does this indicate about the input resistance of the logic gate?

A: this one very much depends on the transistors used to construct the AND gate i.e. its logic family. If the AND gate is a 74LS08 then these transistors have a 'low' input resistance (BJT), so moving to a larger pull down resistor means there is now insufficient current to drive the input to a logic '0' i.e. it is as if the pull-down resistor was removed. If the AND gate is a 74HC08 then these transistors have a 'high' input resistance (CMOS), so moving to a larger pull down resistor still means there is sufficient current to drive a logic '0'. Again think of the input resistance forming a potential divider with the pull-down resistor.

Task 3

A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A	B	Cin	Sum
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Task 4

A	B	Cin	Carry SOP	Sum SOP
0	0	0		
0	0	1		/A*/B*C
0	1	0		/A*B*/C
0	1	1	/A*B*C	
1	0	0		A*/B*/C
1	0	1	A*/B*C	
1	1	0	A*B*/C	
1	1	1	A*B*C	A*B*C

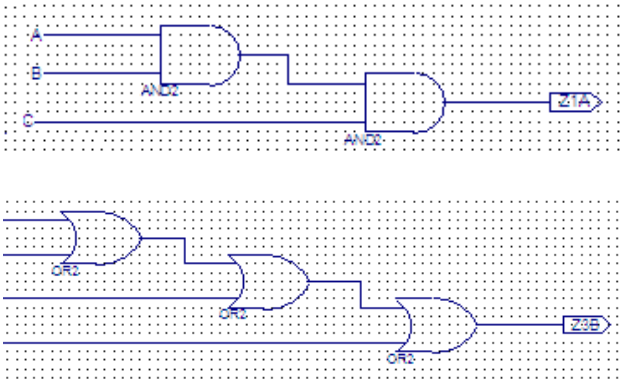
```
CSOP = /A*B*C + A*/B*C + A*B*/C + A*B*C;
SSOP = /A*/B*C + /A*B*/C + A*/B*/C + A*B*C;
```

Task 5

```
CSOP = /A*B*C + A*/B*C + A*B*/C + D;
SSOP = /A*/B*C + /A*B*/C + A*/B*/C + D;
D = A*B*C;
```

NOT gates used to implement `/A`, `/B` and `/C` can also be shared.

2 × Four input OR + 7 × Three input AND + 3 × NOT = 12 gates



2 × 3 × OR + 7 × 2 × AND + 3 × NOT = 23 gates

Sum

Cin \ AB	_00_	_01_	_11_	_10_
_0_		1		1
_1_	1		1	

Cout

Cin \ AB	_00_	_01_	_11_	_10_
_0_			1	
_1_		1	1	1

CSOP = B*C + A*C + A*B;
SSOP = /A*/B*C + /A*B*/C + A*/B*/C + A*B*C;

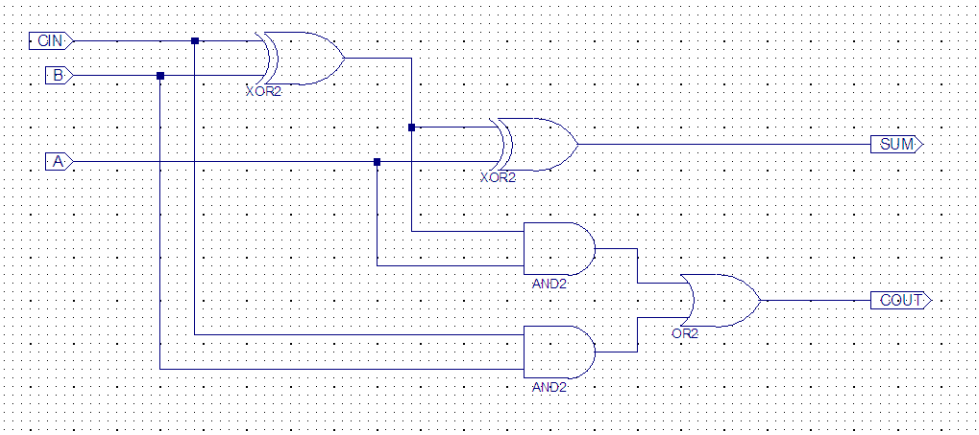
Task 6

```
SSOP = ((/A*B + A*/B) * /C) + ((/A*/B +A*B)*C) ;

SSOP = ((A^B) * /C) + (/(A^B)*C) ;

D = A^B;
SSOP = (D*/C) + (/D*C) ;

SSOP = (D^C) ;
SSOP = ((A^B)^C) ;
SSOP = A^B^C ;
```

```
CSOP = B*C + A*C + A*B;
CSOP = B*C*(A+/A) + A*C*(B+/B) + A*B*(C+/C);
CSOP = A*B*C + /A*B*C + A*B*C + A*/B*C + A*B*C + A*B*/C;
CSOP = (A*B*C + /A*B*C) + (A*/B*C + A*B*/C);
CSOP = (B*C) + (A*(B^C);
```

Task 7

```
CSOP = (B*C) + (A*D);
SSOP = (A^D) ;
D = (B^C);
```



Task 8

As lab script

Task 9

As lab script

Q: the switches contact bounce is simulated by the piecewise linear voltage V2. The circuit shown in figure 14 differs slightly from that shown in figure 12, with the additional of diode D1. Can you see why this component was added.

A: this bypasses the charging resistor allowing the capacitor to quickly return to its default logic '1' state.