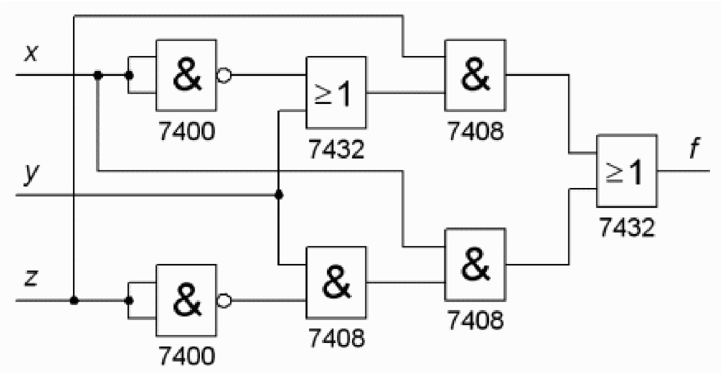
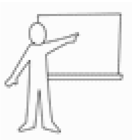
Laboratory combinatorial circuits

Digital Design IE1204/5



Attention! To access the laboratory experiment you must have:

- A booked lab time in the reservation system (Canvas).
- completed your personal knowledge control on the Web (Web-quiz).
- done all preparation tasks mentioned in the lab booklet.



If a preparation task has this "label", you must also be prepared to to present an oral solution for your peers at the lab.

During the lab you work in groups of two, but both students are responsible individually for their preparation and implementation.
Booth students should bring their lab booklets. This frontpage is used as your **receipt** that the lab is completed. Save the receipt until you have received the full course registered in the database (Ladok).

*Since this is your receipt you **must** fill in the table with ink.*

Name:			
Social Security Number:			
• Knowledge Control (Web-quiz)			
Bundle no:		Date:	
Lab-assistant receipt:			
• Preparation tasks in the lab booklet			
Lab-assistant receipt:			
• Lab implementation			
Laboration date:			
Lab-assistant receipt:			

Laboratory combinatorial circuits

Introduction

Breadboard, 74-series

During this lab you will work with breadboards, to be able to easily build, and test a variety of digital networks. The components we use are standard circuits, elementary gates from the so-called 74-series. 74 Series circuits are rarely used in new constructions nowdays, they are mostly considered as "spare parts" for older equipment. The idea here is that the simple gates will give you a concrete picture of what digital technology is all about.
We use chips with "low power" CMOS technology, they only consume power when changing the logic level.

Breadboard is well suited to make easier temporary connections, to test design ideas and to make functional prototypes for school projects or theses. Last in lab booklet is "bill of material" of the equipment, helpful if you ever would need to use simple logic functions yourself.

Simulation of digital circuits

To work with and study basic digital circuits, breadboard and simulator programs are excellent aids. The classic simulation software **Spice** is still widely used by electronics designers and we therefore use a modern variant of this, **LTSpice**, before the laboratory work. Students in the course IE1204 later in the course will use some of the industry's "state of the art" software for digital design and simulation the **QuartusII** and **ModelSim**.

Students in the course IE1205 will instead get these programs demonstrated in the lecture course.

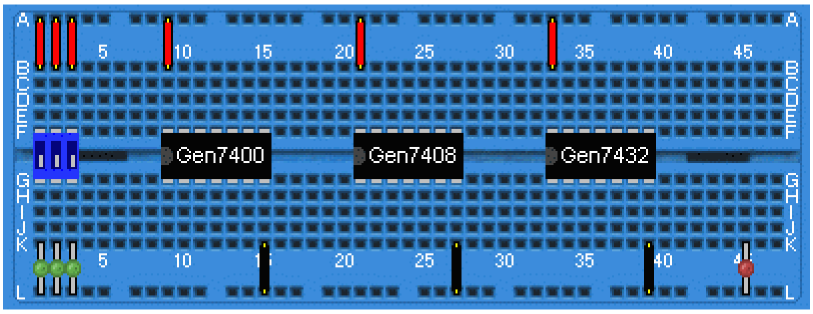
Many of the lab preparatory tasks are to be performed as simulations with **LTSpice**. On course wiki there are links to the tutorial's that show how to do.

The goal of the lab

- Learning how to work with breadboards.
- Orienting yourself on logic functions and circuits.
- Orienting yourself on programs for simulating electronic circuits.
- Show how to enter signals, and how the results are indicated.
- Practice the minimization of logical networks.
- Orienting yourself on combinational function blocks as full adder, multiplexer and decoder.
- Practice addition and subtraction with a Full-adder.

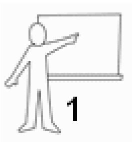
Attention! Your lab time may be prior all course elements that may be needed for the lab has been lectured. You would then have to read the course material for yourself in advance - there are links to all slides for the lectures and exercises.

Truth Table and Boolean expressions



Breadboard with logic gates. An image of the lab equipment.

- ### Preparation task 1 (done before the lab)

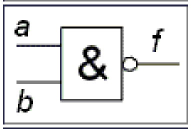


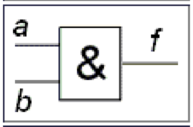
At the lab, you must show that you can use the program **LTSpice** to simulate digital circuits. Start by downloading and installing the program **LTSpice** on your own computet. Follow tutorials on the courseweb.

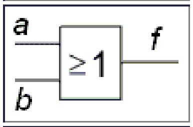
- Simulation result. Here you can enter the value you found that the built-in resistor in our LEDs should have: R = ? [Ω]

Then simulate the truth-tables for the three gates and fill in the simulation result, f_{SIM}, in this booklet.

American Literature, and American symbols are very common. In the figures we use European symbols (which is the international standard), but you should also draw the corresponding American symbols - they are the ones that we use for the simulator-program!

Euro-symbol	USA-symbol
	a b f
7400	7400

Euro-symbol	USA-symbol
	a b f
7408	7408

Euro-symbol	USA-symbol
	a b f
7432	7432

b	a	f_{SIM}	f_{MEAS}
0	0		
0	1		
1	0		
1	1		

Boolean expression $f =$

b	a	f_{SIM}	f_{MEAS}
0	0		
0	1		
1	0		
1	1		

Boolean expression $f =$

b	a	f_{SIM}	f_{MEAS}
0	0		
0	1		
1	0		
1	1		

Boolean expression $f =$

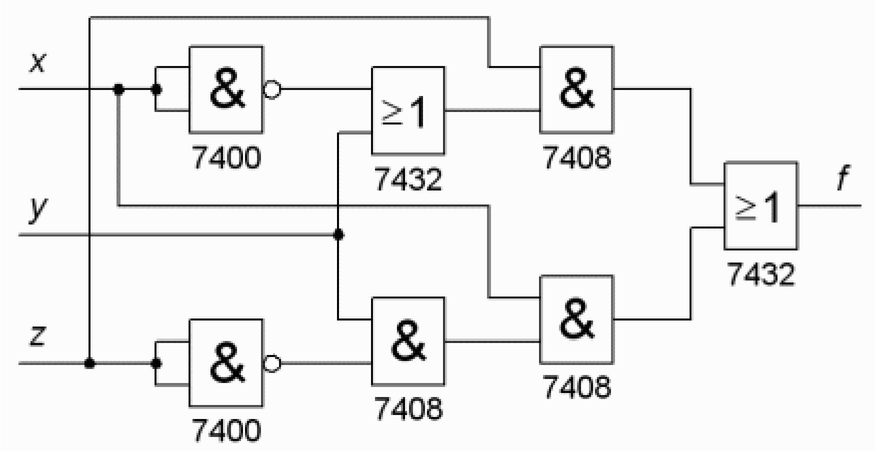
- ### Laboratory task 1

Connect using the soft cables in a similar way as in the simulation. Operate DIL switches with a screwdriver tip to obtain the four input combinations. Measure one gate of each type. Write down the results in the truth tables, f_{MEAS}, and then write the boolean expressions.

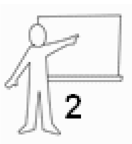
– Attention! You should never "tear down" the basic connections we made in advance on the breadboard. You should just remove the connections you have done yourself.

- You should now have understood how the lab equipment is handled - Do not proceed if the measured values are *not* consistent with the simulated values!

Measurement of a combinational circuit



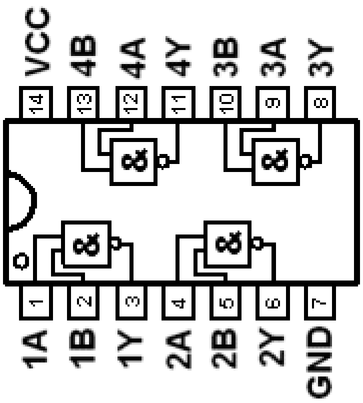
- **Preparation task 2 (done before the lab)**



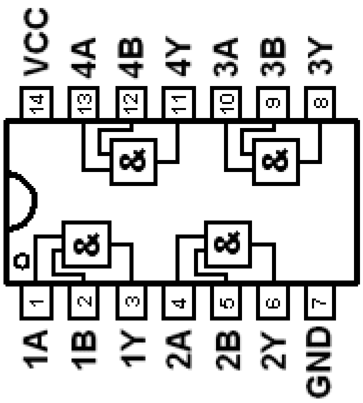
Study the combinatorial network, follow the signals and derive the Boolean equation!

- Boolean expression *f*(*x*, *y*, *z*) =

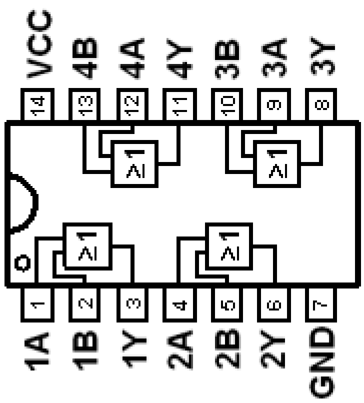
- State all minterms of the function and fill in the truth table, in the column *f*.
- Then connect the circuit in the simulator following the schematic.
- Write down the pin numbers in the schematic in lab booklet of the gates you are using (you choose the gates that will be unused).
- Simulate the circuit all possible input combinations and complete the table with the simulation values, f_{SIM}, in the truth table.



7400



7408



7432

- Try to explain which logic function *f* the circuit realizes? Assume that *x* is a control signal for the circuit. Do you see any pattern/relationship?

- **Laboratory task 2**

Hint! If you know that you have difficulty with this - do then instead Lab Task 3 first, which has fewer connections, before you do this task. With the experience then gained this task becomes easier.

Connect the combinational circuit on a breadboard according to the schematic. You now benefit of having typed the pin numbers. Operate the DIL switches with a screwdriver tip for the 8 different input combinations. Fill in the measurements, f_{MEAS}, in the table.

- Do not proceed if the measured values are *not* consistent with other values!

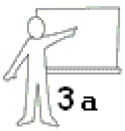
- **Preparation task 3 (done before the lab)**

	x	*y*	*z*	*f*	f_{SIM}	f_{MEAS}	f_{MIN}
0	0	0	0				
1	0	0	1				
2	0	1	0				
3	0	1	1				
4	1	0	0				
5	1	0	1				
6	1	1	0				
7	1	1	1				

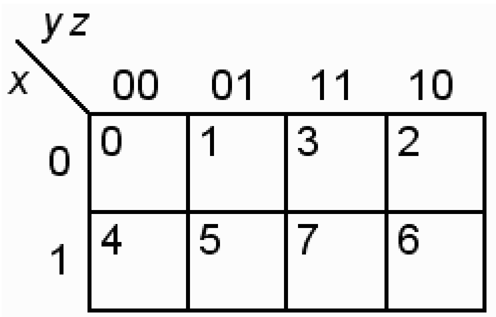
Column f is the truth table of the theoretical values, and column f_{SIM} is for values from simulation in preparation task 2.

Column f_{MEAS} is for measurements from lab task 2.

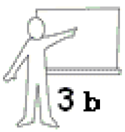
Column f_{MIN} for measurements from the minimized circuit in lab task 3.



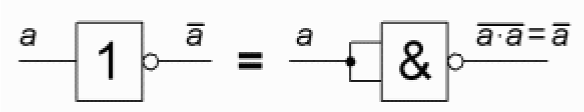
- Transfer the values from the truth table to the Karnaugh map below. Make best groupings and develop the function on the SoP-minimized form (sum of products form).

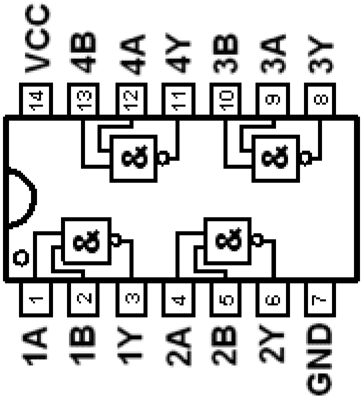


- Minimized Boolean expression $f(x, y, z) =$

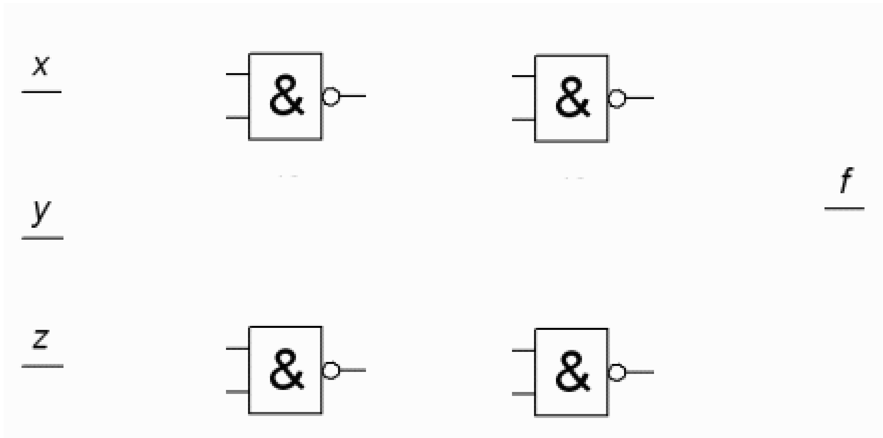


NAND gates are complete logic (all functions can be made using only NAND-gates). Realize the minimized function using only 2-input NAND gates. (See below how to create an inverted variable with a NAND gate.)





- Draw the logic schematic for the minimized function. *Number (pin-number) all used gates inputs and outputs* as the basis for connection of the minimized circuit.



- **Laboratory task 3**

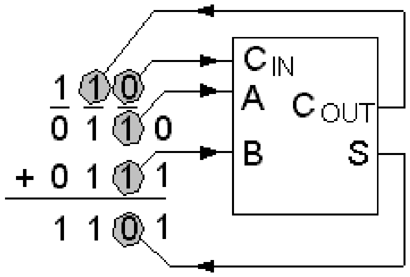
Connect the minimized combinational circuit realized as a NAND-NAND net. Operate DIL switches with a screwdriver tip for the 8 different input combinations. Fill in the measured values, f_{MIN}, in the table.

- Do not proceed if the the measured truth table for the minimized circuit does *not* match with the truth table for the non minimized combinational circuit!

Full adder

- **Preparation task 4 (done before the lab)**

Addition of binary numbers can be done "bit by bit." We call the two bits to be added for A and B. The results, the sum, we call S. If A=B=1 the sum is 1+1=2 (1)0 and for that case there will be a carry. The carry we name C_{OUT}.

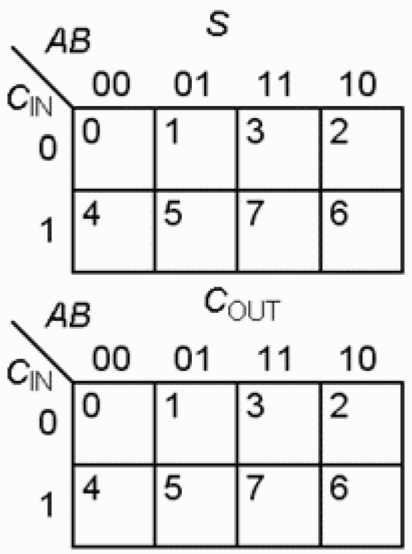


Addition circuit that adds two of the bits in the numbers "6"+"7"="13".

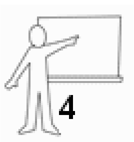
As the figure shows, it can also be a carry C_{IN} from a previous position, it shall then also be included in the addition. Addition circuits must be able to add three bits, and such a circuit is usually called a **full adder**.

- Reasoning your way to the full adder truth table. Fill in columns S and C_{OUT} in the table.

	C_{IN}	A	B	S	S_M	C_{OUT}	C_M
0	0	0	0				
1	0	0	1				
2	0	1	0				
3	0	1	1				
4	1	0	0				
5	1	0	1				
6	1	1	0				
7	1	1	1				



Columns S and C_{OUT} is for your "reasoning" in preparation task 4.
Columns S_M and C_M is for the measurements of the 1-bit full adder in laboratory task 4.

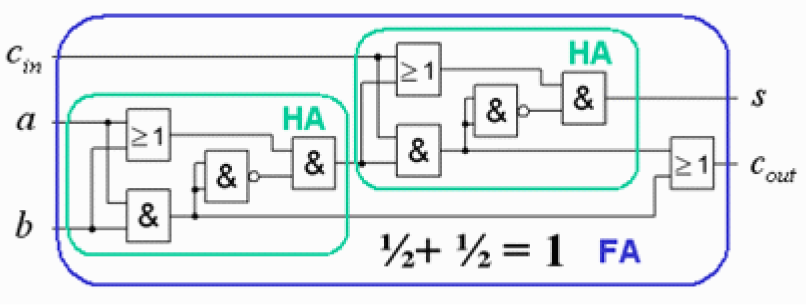


- derive the boolean expressions for the sum bit S and the carry bit C_{OUT}.
- Is it possible to simplify the expressions by using Karnaugh maps?

$S(A, B, C_{IN}) =$

$C_{OUT}(A, B, C_{IN}) =$

Some simulations with ready made simulation files



Two half adders form a full adder (HA+HA=FA). - this circuit is unusual, but it fits the components we have on hand.

You should simulate the Full adder circuit in the schematic above. (A ready made simulation file exists).

- Do *S* and C_{OUT} match your truth table?

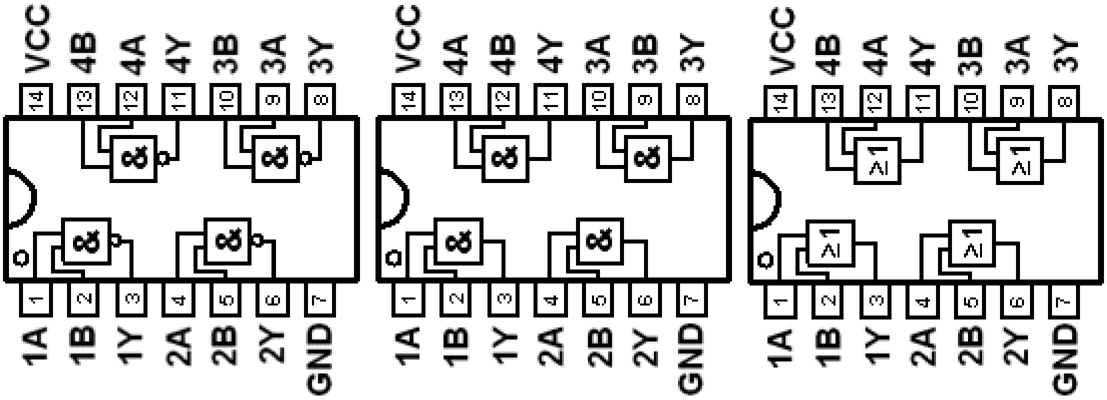
Simulate the full adder circuit with the last full adder stage inside the the 4-bit adder 74283. "Compare" the output signals with two xor-gates (a ready made simulation file exists).

- Are both circuits exactly alike? Take the opportunity to discuss any discrepancies with labassistant later in the lab.

Simulate the two full adder circuits together, but with the inputs to one full adder reversed! (a ready made simulation file exists).

- Is there a difference? What do you conclude? What symmetry properties does the full adder have?

How do you intend to connect full adder circuit in the lab? Write down the pin numbers of the gates you are using in the schematic in lab booklet . It will facilitate the connection work.



- ## Laboratory task 4

Make sure that you have shown your connections with the combinatorial logic on the logic-breadboard for labassistant!

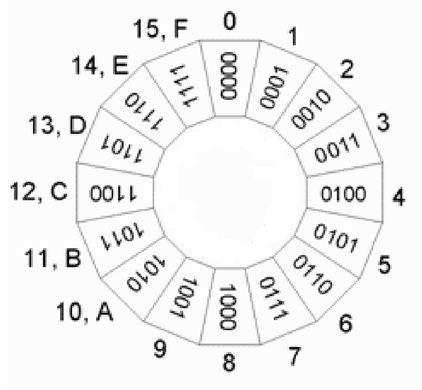
You can then remove all the connections you made with the soft wires.

Now build the full adder circuit on the breadboard. Actuate DIL switches with a screwdriver tip. Insert your readings in the truth table's columns S_M and $_{CM}$.

The idea is that the readings shall correspond to preparation task *S* and $\mathbf{C}_{OUT}$.

Register arithmetic, addition and subtraction

The first microprocessor chip Intel 4004 the year 1971 was a four bit processor, with an embedded fourbit adder as then 74283 we use in the lab. The wordsize four bits is named a **nibble**. With four bits one can express a total of 16 numbers, one hexadecimal digit. The numbersystem is a ring, after 15 one starts from 0 again. See the figure.



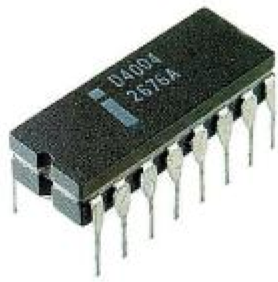
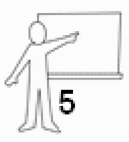


Image of the first microprocessor.

- ## Preparation task 5 (done before the lab)



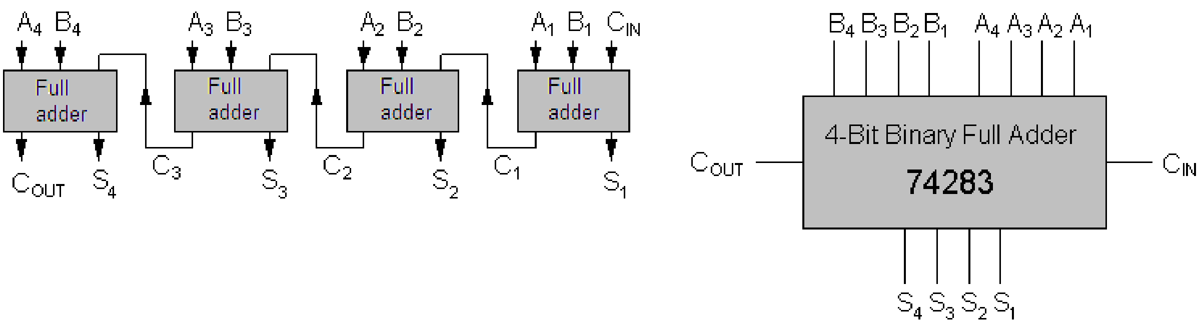
Despite the limited word length still very large numbers can be added by repeated additions!
Follow this addition example with the hexadecimal numbers step by step (paper and pencil, mental arithmetic). If there arises an outgoing Carry, it has to be included as input Carry at the next step.

	(1)	(1)	(1)		**Decimal?**
		F	E	C	
+		1	D	B	
=	1	1	C	7	

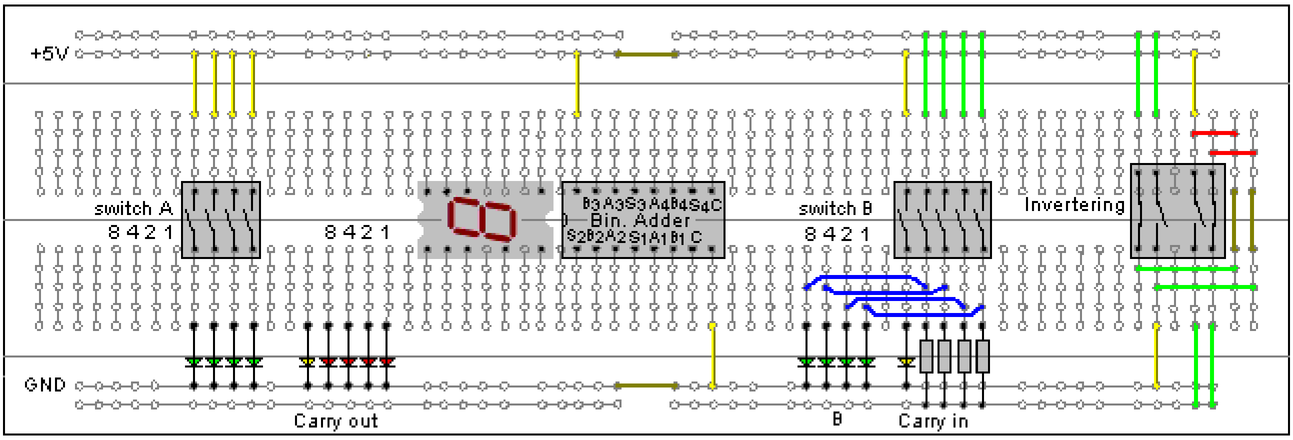
C + B = (1)7, 1+E+D = F+D = (1)C, 1+F+1 = 2 + F = (1)1

- Which decimal numbers corresponds to the hex numbers? Convert the hex numbers to decimal numbers and enter them in the table.
- Then check that the addition is correct with the aid of a calculator.

- ## Laboratory task 5



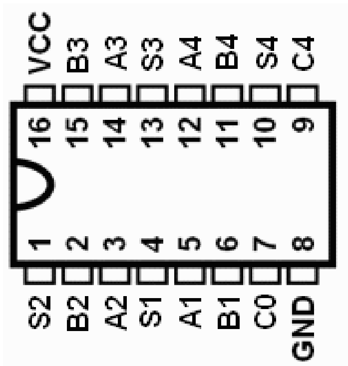
4-bit adder circuit 74283.



Breadboard with a 4-bit adder circuit 74283.

Connections

The chip 74283 is a four bit adder circuit. On the breadboard there are two groups of DIL-switches, switch A and switch B. Tthe signals to the adder inputs are to be taken from these. Connect the input signals from the green lightdiodes to the adder inputs A4 A3 A2 A1 respective B4 B3 B2 B1. The adder C0 (C_{IN}) is to be connected to a "fifth" switch B (with a yellow lightdiode).
The adder outputs S4 S3 S2 S1 is to be connected to the red lightdiodes and the output signal C4 (C_{OUT}) to the adjacent yellow LED.



Possibly there is a HEX display connected to the adder outputs (in this case it is already connected). It can be helpful when you interpret the binary numbers with the LEDs.

- Try the adder with 3 + 3 = ?   (0011+0011)
- What will happen with 8 + 8 = ?   C4 = ?   (1000+1000)
  What information do we get from C4 (C_{OUT})?

The outgoing bit C_{OUT} from a adder is named the **Carry**-flag.

- Use the **adder** to add two multi-digit hexadecimal numbers which you will receive by labassistant. Check that the addition is correct with the aid of a calculator.

+					
=					

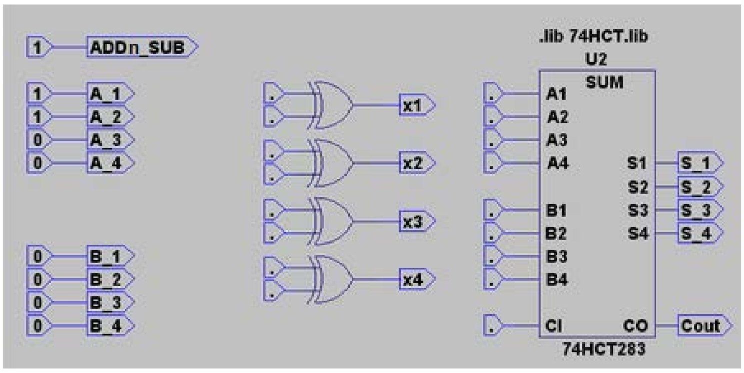
- **Preparation task 6 (done before the lab)**

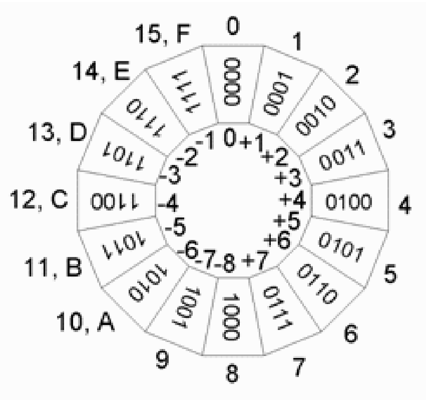
Signed numbers

One can either assume "unsigned numbers" or "signed numbers". When using signed numbers, the number range is divided into two parts, one for the positive numbers and one for the negative. If you back one step from 0 on the number ring you will get $F_{16}=15_{10}=1111_2$ which therefore is the number -1. The border between the positive and the negative numbers is between 8 (-8) and 7 (+7). It is the most significant bit that can be seen as a "signbit". If this bit is 1 the number is negative, it is 0 the number is positive.

You make a number negative by taking the 2-complement of it, wich means that all bits are inverted and that a 1 is added.

A adder circuit can also be used for subtraction if one can take the 2-complement of the subtrahend. Then there are a control signal $\overline{\text{ADD}}/\text{SUB}$ which change the circuit for subtraction. Inverting the bits is used to be done with xor-gates, the adder and its C_{IN} input can then add 1.

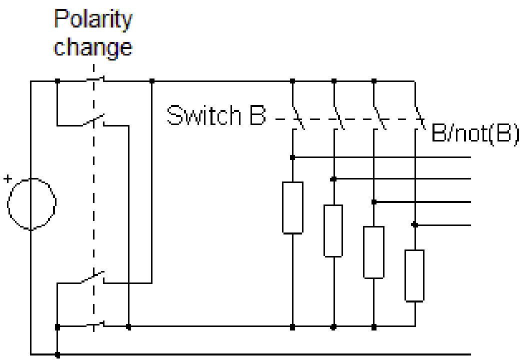




- Think about how the schematic for such a $\overline{\text{ADD}}/\text{SUB}$ circuit would look. Draw and finish in the figure.
- Make negative the number $+3_{10}$ = 0011_2. -3_{10} =

- **Laboratory task 6**

Continue with the adder circuit and keep the connections from before.
On the breadboard there is a polarity changer contact, if you press the button every 1→0 and 0→1 from switchgroup B will be inverted. C0 is operated separately with the fifth switch (yellow lightdiode).
In this way you can make the number from the switch group B negative.



- Try the adder/subtracter with (+3) + (-3) = ?

- wat will happen if we add (+4) + (+4) = ? If we interpret the answer as a signed number.

For operations with signed number, the result may end up on the wrong side of the "sign limit". This is then called **overflow**.

Preparation task 7 (done before the lab)

Multiplication with a constant

Suppose we need to multiply a number *x* to 3.

- This can be done as $2 \cdot x + 1 \cdot x = 3 \cdot x$.

Multiplication with a power of 2, is done by "shifting" the connections of the bits *one* step to the left.

- A more advanced solution uses $2 \cdot (1 \cdot x + 0{,}5 \cdot x) = 3 \cdot x$.

Here the connectors of the bits are shifted both to the left or to the right.

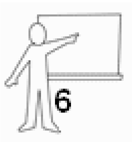
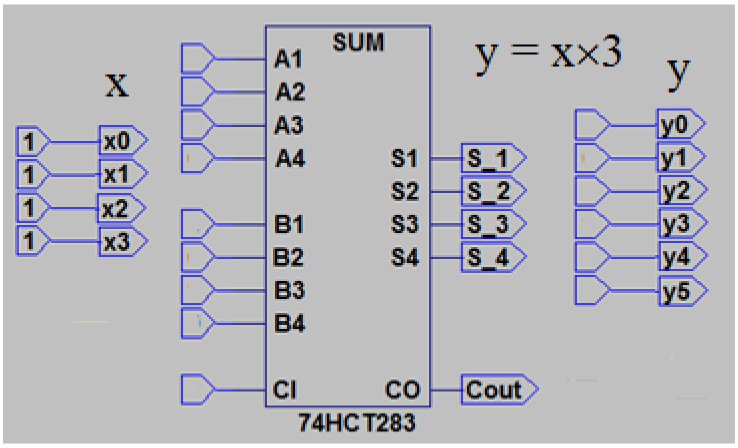


Figure out how an adder can be used as a "multiply by three" circuit. Draw the complete schematic in the figure.



- **Laboratory task 7**

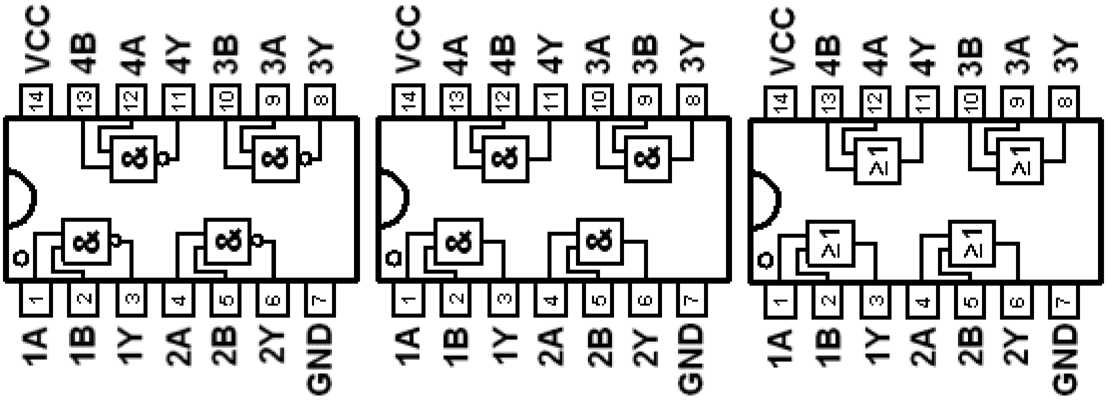
Connect the adder so that it multiplies the number that is set on one switch group with the constant "3". (We now consider unsigned numbers).
(Easiest? Keep the wires from one of the switch groups and reconnect the wires from the other switch group.)

- Do you remember the Multiplication table for the number 3? Set the switch group to 0·3, 1·3, 2·3, 3·3, 4·3, 5·3 6·3 7·3 8·3 …
- Our chip 74283 contains four full adders, can your circuit multiply all 4-bit numbers with 3 (3·15=45)?

Do you have time for more?

Indicate when overflow occurs.
If you are well prepared for the lab, and if you are not suffering from intermittent connections or dead batteries, then you probably now have time for a "voluntary" task.
An optional task may be to make an indicator that "alerts" when an overflow has occurred.
As you noticed, the adder can give wrong results in the addition of signed numbers. The sum of two positive numbers can in addition end up being wrong in the negative range (A_4=0, B_4=0, S_4=1), as the sum of two negative numbers can be wrong and end up in the positive range (A_4=1, B_4=1, S_4=0). A **Overflow-flag** is a bit that warns when this has happened.

$$overflow = A_4 B_4 \overline{S}_4 + \overline{A}_4 \overline{B}_4 S_4$$



Are you a little ingenious you will find that there are enough gates at the lab “logic breadboard” for connecting such an overflow circuit!

- The overflow-circuit can be tested with the DIL-switches and the red lightdiode

Good luck!

When you are finished. Remove all the connections that you have made, but no others, and clean the lab desk.

Bill of materials

The "bill of material" for the lab equipment, could be helpful if you ever would need to use simple logic functions yourself.

```
breadboard GL-12F ELFA 48-427-04
Battery contact ELFA 42-043-01
DIL-switch 3P ELFA 35-395-25
DIL-switch 4P ELFA 35-395-33
DIL-switch 5P ELFA 35-395-41
Lightdiode with series resistor 5V green ELFA 75-014-99
Lightdiode with series resistor 5V red  ELFA 75-012-59
Lightdiode with series resistor 5V yellow  ELFA 75-015-11
NAND gates 74HC0 ELFA 73-500-10
AND gates 74HC08 ELFA 73-503-17
OR gates 74HC32 ELFA 73-510-18
4-bit adder chip 74HC283 ELFA 73-537-82
Pushbutton Double press ELFA 35-650-25
Battery Holder 6V ELFA 4xR6 ELFA 69-506-61
```

Laboratory exercise is compiled by *William Sandqvist* william@kth.se