

# **Experiment 6: Design and Implementation of a Full Subtractor using Basic Gates and the NAND Universal Gate**

Peter Dranishnikov

U0000005258

EEL3702C

Digital Logic Design

Section 04

Professor: Dr. Balasubramaniyan Chandrasekaran

## **Objectives:**

This experiment involved the design method of a combinational digital circuit to formulate the boolean function and the equivalent physical circuit of a full subtractor. Additionally, the circuit was redesigned using only one universal gate: the NAND gate.

## **Experimental Equipment:**

Q	Identifier	Item Description
1	Keithley 3	5V Power Supply
2	SN74LS08N	2-input TTL AND Gate Package
2	SN74LS04N	TTL Inverter Package
2	SN74LS32N	2-input TTL OR Gate Package
5	SN74LS00N	2-input TTL NAND Gate Package
1	SN74LS20N	4-input TTL NAND Gate Package
1	SN74LS86AN	2-input XOR Gate Package
2	GREEN	LED
4	330 $\Omega$	Resistor (5%)
2	RED	LED
1	Global Solutions	Breadboard
?	28AWG	Connector Wires

## **Description of Procedural Methods:**

### 1. Problem Definition:

1. A full subtractor consists of the difference between minuend, the subtrahend, and a borrow from the less significant digit, in that order. A borrow-out consists of boolean logic regarding the comparison of the minuend and the combined sum of the subtrahend and the borrow-in; if the sum is greater than the minuend, then the borrow-out is boolean 1, else 0.

### 2. Truth Table

X	Y	$B_i$	D	$B_o$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0

<b>X</b>	<b>Y</b>	<b>B<sub>i</sub></b>	<b>D</b>	<b>B<sub>o</sub></b>
1	1	0	0	0
1	1	1	1	1

### 3. Boolean Function

#### 1. K-Map Simplification:

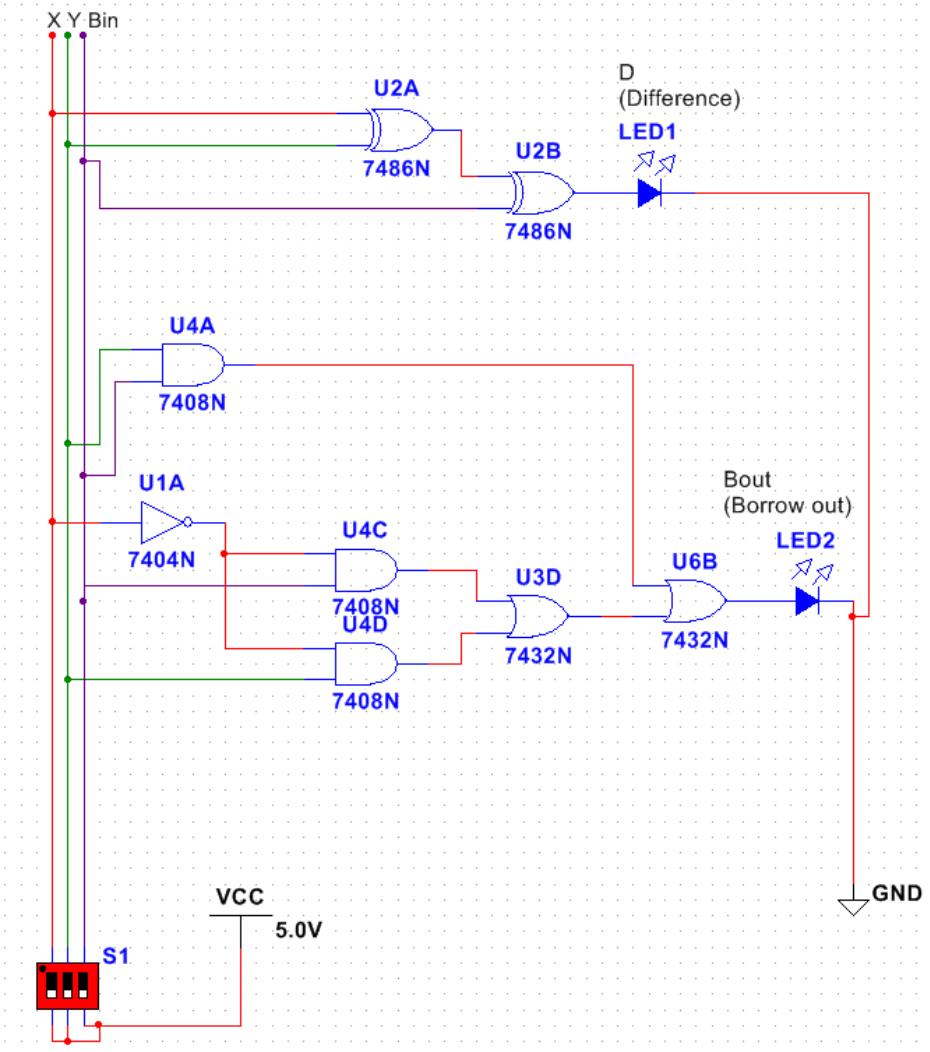
D	Y'B <sub>i</sub> '	Y'B <sub>i</sub>	YB <sub>i</sub>	YB <sub>i</sub> '
X'	0	1	0	1
X	1	0	1	0
B <sub>o</sub>	Y'B <sub>i</sub> '	Y'B <sub>i</sub>	YB <sub>i</sub>	YB <sub>i</sub> '
X'	0	1	1	1
X	0	0	1	0

#### 2. Formulation

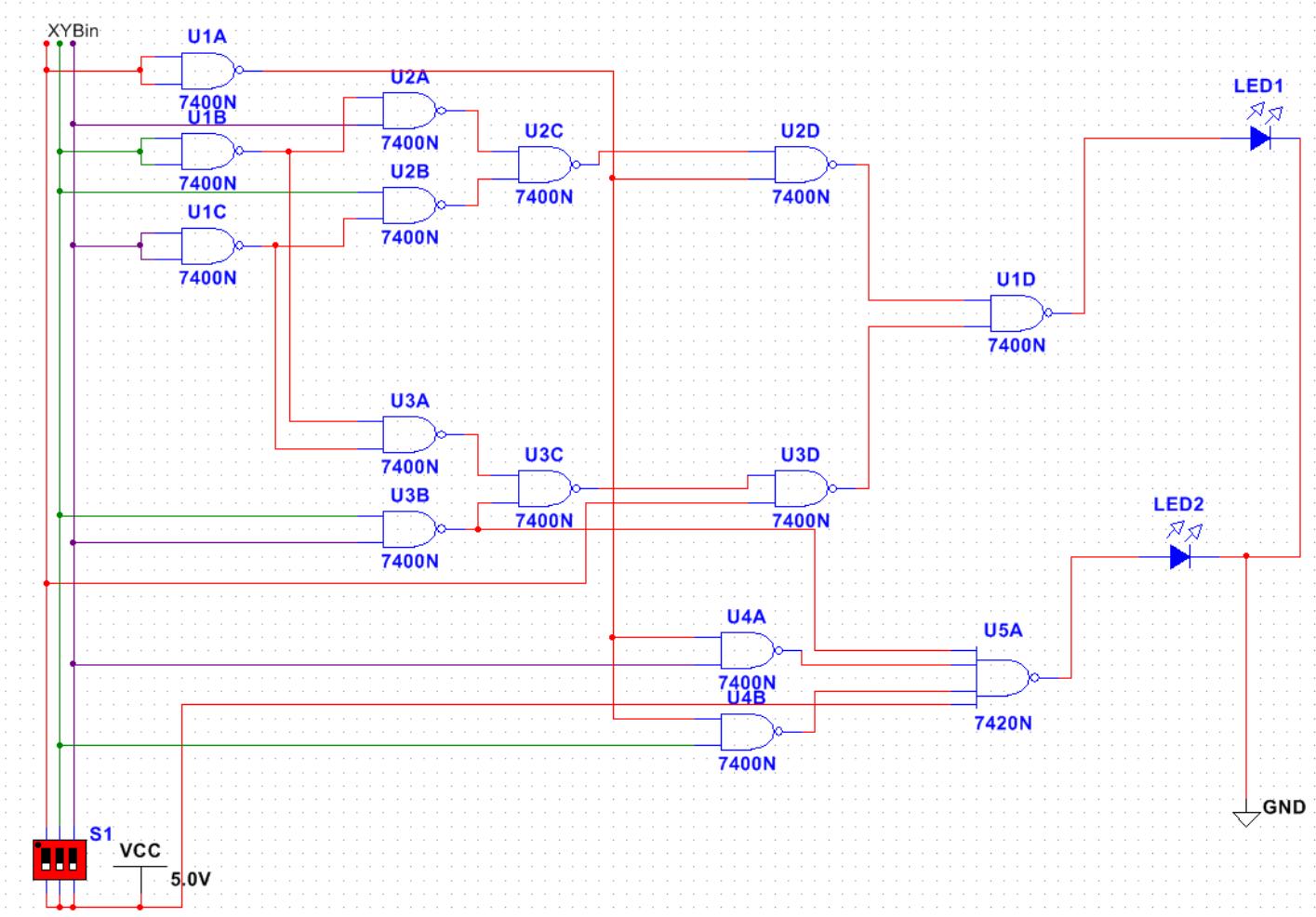
1.  $D = X'Y'B_i + X'YB_i' + XY'B_i' + XYB_i$ 
  1. Alternatively:  $D = X'(Y'B_i + YB_i') + X(Y'B_i' + YB_i)$
  2.  $B_o = X'B_i + X'Y + YB_i$

#### 4. Circuit implementation

##### 1. AND-OR-Invert (schematic includes XOR for simplicity)



## 2. NAND Only



### 5. Physical implementation:

1. See check off sheet for results on next page
2. Experimental truth table:

X	Y	B <sub>i</sub>	D	B <sub>o</sub>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

**EEL3702C**  
**Lab Check off Sheet**

Please bring a print out of this sheet before demonstrating your working circuit. Attach this check off sheet with the Lab report.

*Note: Lab check off sheet should be signed by the Instructor or TA. Lab reports without the check off sheet will not receive credit.*

**Name:** Peter Dranishnikov

**Section:** 04

**Experiment:** 6

Lab Demonstration Comments:

Basic B W 03/02

Nand PW

Signature of Instructor/TA:

Date:

## **Results/Discussion:**

The original AND-OR-Invert implementation initially did not use XOR, but the professor stated that it was allowed, so the implementation was simplified with XOR gates for the difference output. The initial NAND implementation (not shown) failed to function due to incorrect gate layering after bubble pushing, so a redesign was required (and verified outside of allotted lab time).

Both designs presented in this report are functional in simulation and implementation, as verified by the checkoff sheet and the matching truth table.

## **Conclusion:**

Given a desired functionality with input parameters, some amount of wires, gates, a method of connection (breadboard, solder), and patience/time, it is possible to design and implement any combinational digital logic circuit using the digital logic design process.