

LOGISIM-EVOLUTION LAB MANUAL

GEORGE SELF

July 2019 – Edition 4.0

George Self: *Logisim-Evaluation Lab Manual*,

This work is licensed under a **Creative Commons** “CCo 1.0 Universal” license.



PREFACE

I have taught CIS 221, *Digital Logic*, for Cochise College since about 2003 and enjoy working with students on this topic. From the start, I wanted students to work with labs as part of our studies and actually design circuits to complement our theoretical instruction. As I evaluated circuit design software I had three criteria:

- **Open Educational Resource (OER).** It is important to me that students use software that is available free of charge and is supported by the entire web community.
- **Platform.** While most of my students use a Windows-based system, some use Macintosh and it was important to me to use software that is available for both of those platforms. As a bonus, most OER software is also available for the Linux system, though I'm not aware of any of my students who are using Linux.
- **Simplicity.** I wanted to use software that was easy to master so students could spend their time understanding digital logic rather than learning the arcane structures of a simulation language.

I originally wrote a number of lab exercises using *Logisim*, but the creator of that software, Carl Burch, announced that he would quit developing it in 2014. Because it was published as an open source project, a group of Swiss institutes started with the *Logisim* software and developed a new version that integrated several new tools, like a chronogram, and released it under the name *Logisim-Evolution*.

It is my hope that students will find these labs instructive and they will enhance their learning of digital logic. This lab manual is written with \LaTeX and published under a [Creative Commons Zero](#) license with a goal that other instructors can modify it to meet their own needs. The source code can be found at [my personal GITHUB page](#) and I always welcome comments that will help me improve this manual.

—George Self

BRIEF CONTENTS

List of Figures ix

List of Tables ix

Listings xi

I INTRODUCTION TO LOGISIM-EVOLUTION 1

II FOUNDATIONS 3

III COMBINATIONAL CIRCUITS 5

1 ARITHMETIC LOGIC UNIT (ALU) 7

IV SEQUENTIAL CIRCUITS 13

V SIMULATION 15

VI APPENDIX 17

A TTL REFERENCE 19

CONTENTS

List of Figures ix

List of Tables ix

Listings xi

I INTRODUCTION TO LOGISIM-EVOLUTION 1

II FOUNDATIONS 3

III COMBINATIONAL CIRCUITS 5

1 ARITHMETIC LOGIC UNIT (ALU) 7

1.1 Purpose 7

1.2 Procedure 8

1.2.1 main 8

1.2.2 ALU 8

1.2.3 Arithmetic 9

1.2.4 Challenge 10

1.2.5 Testing the Circuit 11

1.3 Deliverable 11

IV SEQUENTIAL CIRCUITS 13

V SIMULATION 15

VI APPENDIX 17

A TTL REFERENCE 19

A.1 7400: Quad 2-Input NAND Gate 19

A.2 7402: Quad 2-Input NOR Gate 20

A.3 7404: Hex Inverter 21

A.4 7408: Quad 2-Input AND Gate 22

A.5 7410: Triple 3-Input NAND Gate 23

A.6 7411: Triple 3-Input AND Gate 24

A.7 7413: Dual 4-Input NAND Gate (Schmitt-Trigger) 25

A.8 7414: Hex Inverter (Schmitt-Trigger) 26

A.9 7418: Dual 4-Input NAND Gate (Schmitt-Trigger Inputs) 27

A.10 7419: Hex Inverter (Schmitt-Trigger) 28

A.11 7420: Dual 4-Input NAND Gate 29

A.12 7421: Dual 4-Input AND Gate 30

A.13 7424: Quad 2-Input NAND Gate (Schmitt-Trigger) 31

A.14 7427: Triple 3-Input NOR Gate 32

A.15 7430: Single 8-Input NAND Gate 33

A.16 7432: Quad 2-Input OR Gate 34

A.17 7436: Quad 2-Input NOR Gate 35

A.18	7442: BCD to Decimal Decoder	36
A.19	7443: Excess-3 to Decimal Decoder	37
A.20	7444: Gray to Decimal Decoder	39
A.21	7447: BCD to 7-Segment Decoder	41
A.22	7451: Dual AND-OR-INVERT Gate	43
A.23	7454: Four Wide AND-OR-INVERT Gate	44
A.24	7458: Dual AND-OR Gate	45
A.25	7464: 4-2-3-2 AND-OR-INVERT Gate	46
A.26	7474: Dual D-Flipflops with Preset and Clear	47
A.27	7485: 4-Bit Magnitude Comparator	48
A.28	7486: Quad 2-Input XOR Gate	48
A.29	74125: Quad Bus Buffer, 3-State Gate	49
A.30	74165: 8-Bit Parallel-to-Serial Shift Register	50
A.31	74175: Quad D-Flipflops with Sync Reset	51
A.32	74266: Quad 2-Input XNOR Gate	51
A.33	74273: Octal D-Flipflop with Clear	52
A.34	74283: 4-Bit Binary Full Adder	53
A.35	74377: Octal D-Flipflop with Enable	54

LIST OF FIGURES

Figure 1.1	ALU main	8
Figure 1.2	ALU Subcircuit	9
Figure 1.3	Arithmetic Subcircuit	10
Figure 1.4	Logic Subcircuit	10
Figure A.1	Three Surface-Mounted Integrated Circuits	19
Figure A.2	7400: Single NAND Gate Circuit	19
Figure A.3	7402: Single NOR Gate Circuit	20
Figure A.4	7404: Single Inverter Circuit	21
Figure A.5	7408: Single AND Gate Circuit	22
Figure A.6	7410: Single 3-Input NAND Gate Circuit	23
Figure A.7	7411: Single 3-Input AND Gate Circuit	24
Figure A.8	7413: Single 4-Input NAND Gate Circuit	25
Figure A.9	7414: Single Inverter Circuit	26
Figure A.10	7418: Single 4-Input NAND Gate Circuit	27
Figure A.11	7419: Single Inverter Circuit	28
Figure A.12	7420: Single 4-Input NAND Gate Circuit	29
Figure A.13	7421: Single 4-Input AND Gate Circuit	30
Figure A.14	7424: Single NAND Gate Circuit	31
Figure A.15	7411: Single 3-Input NOR Gate Circuit	32
Figure A.16	7430: Single 8-Input NAND Gate	33
Figure A.17	7432: Single OR Gate Circuit	34
Figure A.18	7436: Single NOR Gate Circuit	35
Figure A.19	7442: BCD to Decimal Decoder	36
Figure A.20	7447: BCD to 7-Segment Decoder	41
Figure A.21	7451: Single AND-OR-INVERT Gate Circuit	43
Figure A.22	7454: Four Wide AND-OR-INVERT Gate Circuit	44
Figure A.23	7458: Dual AND-OR Gate Circuit	45
Figure A.24	7464: 4-2-3-2 AND-OR-INVERT Gate Circuit	46
Figure A.25	7486: Single XOR Gate Circuit	48
Figure A.26	74125: Single Buffer Circuit	49
Figure A.27	74266: Single XNOR Gate Circuit	51

LIST OF TABLES

Table 1.1	Function Table for 74181 ALU	7
Table A.1	Pinout For 7400	20
Table A.2	Pinout For 7402	21

Table A.3	Pinout For 7404	22	
Table A.4	Pinout For 7408	23	
Table A.5	Pinout For 7410	24	
Table A.6	Pinout For 7411	25	
Table A.7	Pinout For 7413	26	
Table A.8	Pinout For 7414	27	
Table A.9	Pinout For 7418	28	
Table A.10	Pinout For 7419	29	
Table A.11	Pinout For 7420	30	
Table A.12	Pinout For 7421	31	
Table A.13	Pinout For 7424	32	
Table A.14	Pinout For 7427	33	
Table A.15	Pinout For 7430	34	
Table A.16	Pinout For 7432	35	
Table A.17	Pinout For 7436	36	
Table A.18	Truth Table For The 7442 Circuit		37
Table A.19	Pinout For 7442	37	
Table A.20	Truth Table For The 7443 Circuit		38
Table A.21	Pinout For 7443	39	
Table A.22	Truth Table For The 7444 Circuit		40
Table A.23	Pinout For 7444	40	
Table A.24	Truth Table For The 7447 Circuit		42
Table A.25	Pinout For 7447	43	
Table A.26	Pinout For 7451	44	
Table A.27	Pinout For 7454	45	
Table A.28	Pinout For 7458	46	
Table A.29	Pinout For 7464	47	
Table A.30	Pinout For 7474	47	
Table A.31	Pinout For 7485	48	
Table A.32	Pinout For 7486	49	
Table A.33	Pinout For 74125	50	
Table A.34	Pinout For 74165	50	
Table A.35	Pinout For 74175	51	
Table A.36	Pinout For 74266	52	
Table A.37	Pinout For 74273	53	
Table A.38	Pinout For 74283	54	
Table A.39	Pinout For 74377	55	

LISTINGS

ACRONYMS

ALU	Arithmetic Logic Unit
IC	Integrated Circuit
OER	Open Educational Resource
TTL	Transistor-Transistor Logic

Part I

INTRODUCTION TO LOGISIM-EVOLUTION

LOGISIM-EVOLUTION is used to create and test simulations of digital circuits. This part of the lab manual includes only one lab designed to introduce *Logisim-evolution* and teach the fundamentals of using this application.

Part II

FOUNDATIONS

FOUNDATIONAL EXERCISES are designed to provide practice with simple logic circuits in order to both develop skill with *Logisim-evolution* and illustrate the foundations of digital logic.

Part III

COMBINATIONAL CIRCUITS

COMBINATIONAL LOGIC is the bedrock for all digital logic circuits. A combinational circuit's output is determined only by the status of the various inputs and an external clock signal is not necessary as in sequential circuits. All of the circuits completed so far in this manual have been combinational and the two labs in this part of the manual are designed to further develop the concepts of combinational digital logic with two relatively complex examples.

ARITHMETIC LOGIC UNIT (ALU)

1.1 PURPOSE

In this lab you will build an Arithmetic Logic Unit (ALU). An ALU is an important digital logic device used to perform all sorts of arithmetic and logic functions in a circuit. The commercial 74181 ALU has two four-bit data inputs along with a one-bit mode (M) and a four-bit select input. Depending on those settings, the device will complete one of the functions listed in Table 1.1.

Select	Logic (M=1)	Arithmetic (M=0)
0000	A'	A
0001	$(A + B)'$	$A + B$
0010	$A'B$	$A + B'$
0011	Logical 0	minus 1 (2's Comp)
0100	$(AB)'$	$A + AB'$
0101	B'	$(A + B)$ plus AB'
0110	$A \text{ XOR } B$	A minus B minus 1
0111	AB'	AB' minus 1
1000	$A' + B$	A plus AB
1001	$(A \text{ XOR } B)'$	A plus B
1010	B	$(A + B')$ plus AB
1011	AB	AB minus 1
1100	Logical 1	A plus A
1101	$A + B'$	$(A + B)$ plus A
1110	$A + B$	$(A + B')$ plus A
1111	A	A minus 1

Table 1.1: Function Table for 74181 ALU

Notes: in the "Arithmetic" column, the + sign indicates logic OR while the words *plus* and *minus* indicate arithmetic add and subtract operations. The value of A plus A is the same as shifting the bits left to the next most significant position.

The ALU built in this lab is not as complex as a 74181 Integrated Circuit (IC), however it demonstrates the basic functions of an ALU.

1.2 PROCEDURE

This is a rather complex circuit so several completed subcircuits are provided.

Load the **ALU** starter circuit in *Logisim-evolution*. That starter circuit already has the **main**, **ALU**, and **Arithmetic** subcircuits completed.

1.2.1 *main*

The **main** circuit does nothing more than provide a human-friendly interface for the rest of the **ALU**. That interface include two four-bit inputs (labeled *InA* and *InB*), a three-bit select, a one-bit mode, a carry-in and carry-out bit (so the **ALU** could be chained to another to create an eight-bit device), a *compare* output (TRUE if the two inputs are equal), and a four-bit output (labeled *ALUOut*). In operation, numbers are entered on *InA* and *InB*, the mode and select are set, and then the result is read on *ALUOut*.

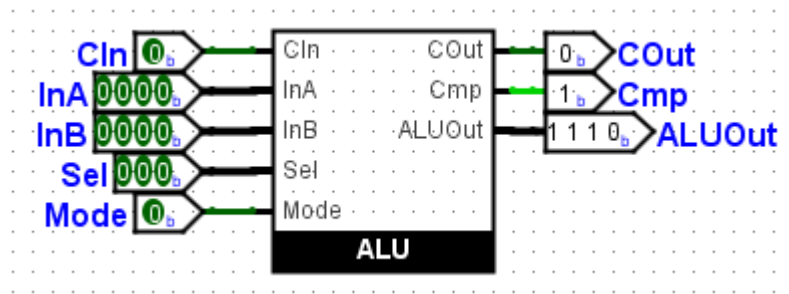


Figure 1.1: ALU main

1.2.2 *ALU*

The **ALU** subcircuit contains the logic that routes *InA*, *InB*, and *Sel* to two other subcircuits, **Arithmetic** or **Logic**. It then uses a multiplexer to route the output of one of those subcircuits to an output port depending on the setting of the *Mode* bit. Note that the inputs are sent to both subcircuits but only the output specified by the *Mode* is returned to the user. This type of logic is also used in the **Arithmetic** circuit.

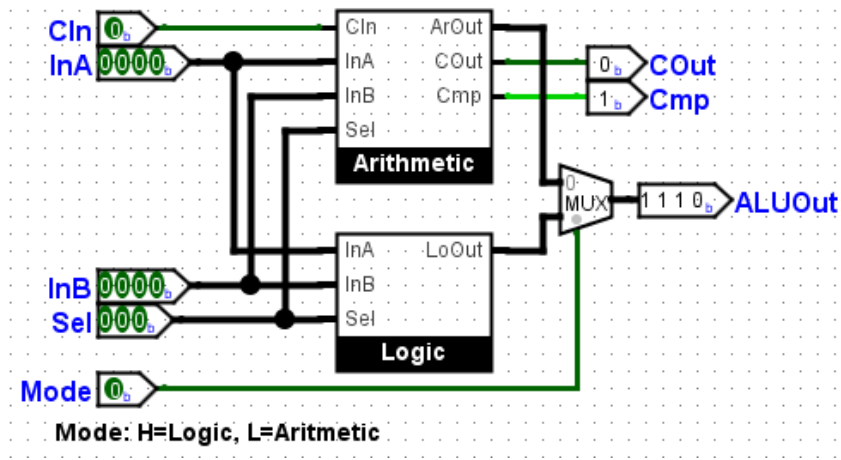


Figure 1.2: ALU Subcircuit

1.2.3 Arithmetic

This subcircuit contains numerous devices from the *Arithmetic* library and they are all wired appropriately for whatever operation is selected. The concept for this subcircuit is rather simple but routing the wiring to all of the devices is challenging.

Notice that two multiplexers are necessary since the circuit provides two different outputs. The top multiplexer routes the four-bit solution and the bottom multiplexer routes the carry-out bit. The *compare* output is always active since it is comparing the input signals and does not rely on the function that is selected.

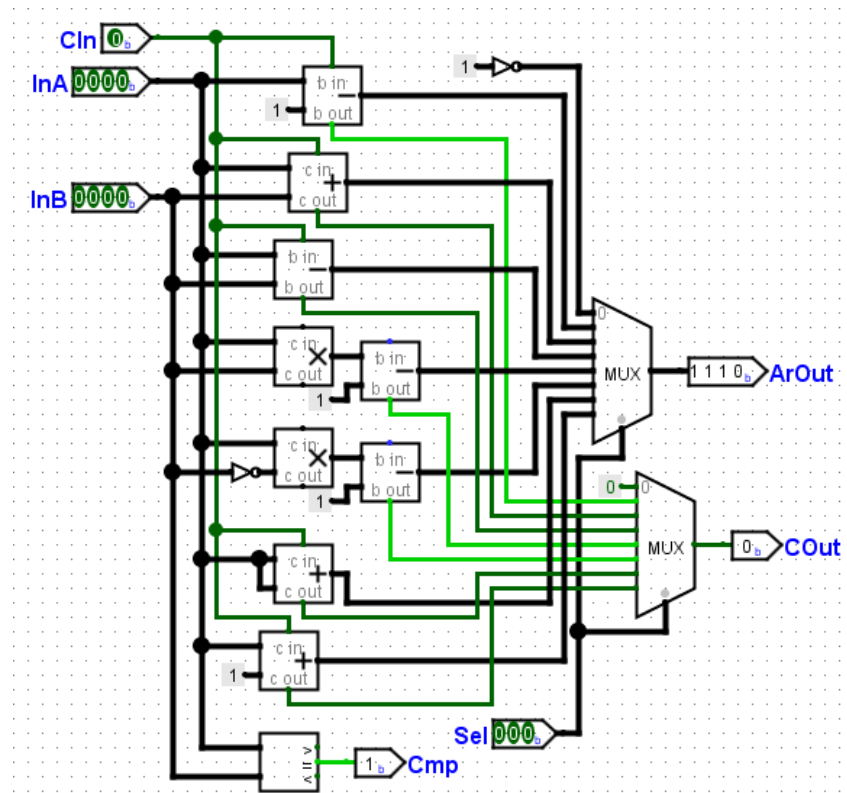


Figure 1.3: Arithmetic Subcircuit

1.2.4 Challenge

In the starter circuit, the **Logic** subcircuit is only a shell with three inputs and one output.

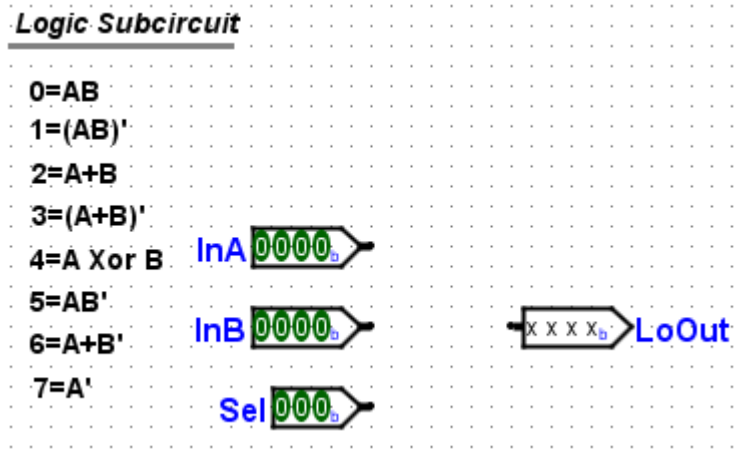


Figure 1.4: Logic Subcircuit

Complete that subcircuit by adding the necessary logic gates and wiring, similar to the **Arithmetic** subcircuit. This subcircuit is much

simpler than the **Arithmetic** subcircuit since there are no carry-in, carry-out, or compare bits. When completed, the subcircuit only needs eight logic gates and a multiplexer added to the starter.

1.2.5 *Testing the Circuit*

The **ALU** should be tested by entering several values on *InA* and *InB* and then select all possible arithmetic and logic operations. The outputs for each check should be accurate.

1.3 DELIVERABLE

To receive a grade for this lab, complete the Challenge. Be sure the standard identifying information is at the top left of the *main* circuit, similar to this:

George Self
Lab 04: ALU
February 18, 2018

Save the file with this name: *Lab04_ALU* and submit that file for grading.

Part IV

SEQUENTIAL CIRCUITS

SEQUENTIAL LOGIC circuits develop the concepts of clock-driven logic while creating several practical counters and memory circuits. These labs also introduce the *Logisim-evolution Chronogram*, which builds timing diagrams for sequential logic circuits.

Part V

SIMULATION

SIMULATION is the most complex topic covered in this lab manual. Included in this manual are a simple processor, designed to teach the foundations of a Central Processing Unit, and an elevator simulator, designed to be a capstone project.

Part VI

APPENDIX

TTL REFERENCE

Logisim-Evolution includes a number of Transistor-Transistor Logic (TTL) ICs. These are pre-packaged digital logic circuits that perform specific, well-defined functions. There are, literally, hundreds of TTL ICs available for purchase from electronics warehouses but *Logisim-Evolution* includes only 35 of the most commonly-used devices. Figure A.1 shows three surface-mounted ICs on a circuit board.

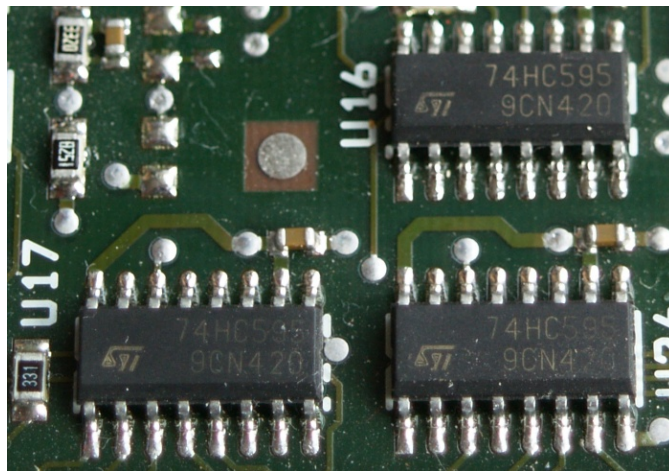


Figure A.1: Three Surface-Mounted Integrated Circuits

A.1 7400: QUAD 2-INPUT NAND GATE

This device contains four independent 2-input NAND gates. Figure A.2 is a logic diagram of one of the four circuits.

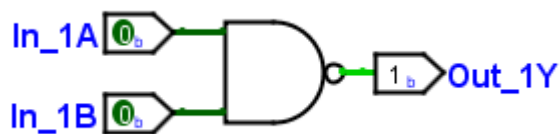


Figure A.2: 7400: Single NAND Gate Circuit

The 7400 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.1.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.1: Pinout For 7400

A.2 7402: QUAD 2-INPUT NOR GATE

This device contains four independent 2-input NOR gates. Figure A.3 is a logic diagram of one of the four circuits.

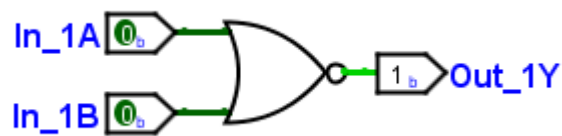


Figure A.3: 7402: Single NOR Gate Circuit

The 7402 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.2.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.2: Pinout For 7402

A.3 7404: HEX INVERTER

This device contains six independent inverters. Figure A.4 is a logic diagram of one of the six circuits.

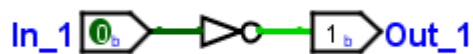


Figure A.4: 7404: Single Inverter Circuit

The 7404 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.3.

Logisim Label	Function
Input: 1	In 1
Output: 2	Out 1
Input: 3	In 2
Output: 4	Out 2
Input: 5	In 3
Output: 6	Out 3
Output: 8	Out 4
Input: 9	In 4
Output: 10	Out 5
Input: 11	In 5
Output: 12	Out 6
Input: 13	In 6

Table A.3: Pinout For 7404

A.4 7408: QUAD 2-INPUT AND GATE

This device contains four independent 2-input AND gates. Figure A.5 is a logic diagram of one of the four circuits.

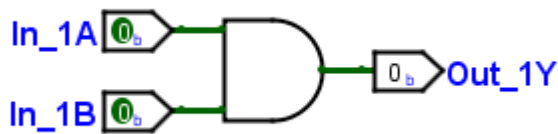


Figure A.5: 7408: Single AND Gate Circuit

The 7408 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.4.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.4: Pinout For 7408

A.5 7410: TRIPLE 3-INPUT NAND GATE

This device contains three independent 3-input NAND gates. Figure A.6 is a logic diagram of one of the three circuits.

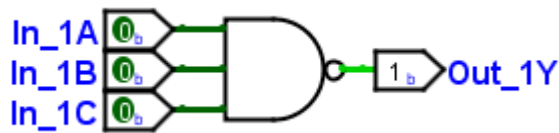


Figure A.6: 7410: Single 3-Input NAND Gate Circuit

The 7410 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.5.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Input: 3	In 2A
Input: 4	In 2B
Input: 5	In 2C
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Input: 11	In 3C
Output: 12	Out 1Y
Input: 13	In 1C

Table A.5: Pinout For 7410

A.6 7411: TRIPLE 3-INPUT AND GATE

This device contains three independent 3-input AND gates. Figure A.7 is a logic diagram of one of the three circuits.



Figure A.7: 7411: Single 3-Input AND Gate Circuit

The 7411 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.6.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Input: 3	In 2A
Input: 4	In 2B
Input: 5	In 2C
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Input: 11	In 3C
Output: 12	Out 1Y
Input: 13	In 1C

Table A.6: Pinout For 7411

A.7 7413: DUAL 4-INPUT NAND GATE (SCHMITT-TRIGGER)

This device contains two independent 4-input NAND gates. Schmitt-triggers are a special type of device that are used to filter out spurious noise on a circuit. They are designed to change from low-to-high or high-to-low only when the input voltage reaches a preset level but not if the voltage randomly fluctuates without crossing the set-points. This device is essentially the same as the 7418. Figure A.8 is a logic diagram of one of the two circuits.

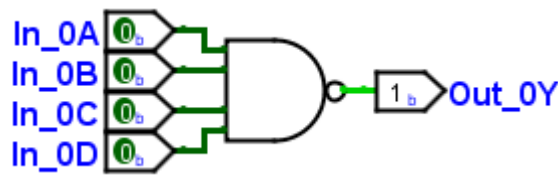


Figure A.8: 7413: Single 4-Input NAND Gate Circuit

The 7413 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.7.

Logisim Label	Function
Input: 1	In A0
Input: 2	In B0
Pin 3: NC	Not Connected
Input: 4	In C0
Input: 5	In D0
Output: 6	Out Y0
Output: 8	Out Y1
Input: 9	In D1
Input: 10	In C1
Pin 11: NC	Not Connected
Input: 12	In B1
Input: 13	In A1

Table A.7: Pinout For 7413

A.8 7414: HEX INVERTER (SCHMITT-TRIGGER)

This device contains six independent inverters. Schmitt-triggers are a special type of device that are used to filter out spurious noise on a circuit. They are designed to change from low-to-high or high-to-low only when the input voltage reaches a preset level but not if the voltage randomly fluctuates without crossing the set-points. This device is essentially the same as the 7419. Figure A.9 is a logic diagram of one of the six circuits.

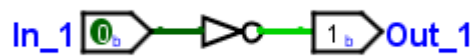


Figure A.9: 7414: Single Inverter Circuit

The 7414 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.8.

Logisim Label	Function
Input: 1	In 1
Output: 2	Out 1
Input: 3	In 2
Output: 4	Out 2
Input: 5	In 3
Output: 6	Out 3
Output: 8	Out 4
Input: 9	In 4
Output: 10	Out 5
Input: 11	In 5
Output: 12	Out 6
Input: 13	In 6

Table A.8: Pinout For 7414

A.9 7418: DUAL 4-INPUT NAND GATE (SCHMITT-TRIGGER INPUTS)

This device contains two independent 4-input NAND gates. Schmitt-triggers are a special type of device that are used to filter out spurious noise on a circuit. They are designed to change from low-to-high or high-to-low only when the input voltage reaches a preset level but not if the voltage randomly fluctuates without crossing the set-points. This device is essentially the same as the 7413. Figure A.10 is a logic diagram of one of the two circuits.

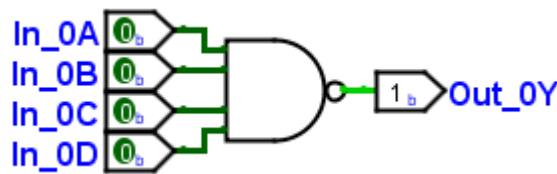


Figure A.10: 7418: Single 4-Input NAND Gate Circuit

The 7418 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.9.

Logisim Label	Function
Input: 1	In A0
Input: 2	In B0
Pin 3 NC	Not Connected
Input: 4	In C0
Input: 5	In D0
Output: 6	Out Y0
Output: 8	Out Y1
Input: 9	In D1
Input: 10	In C1
Pin 11 NC	Not Connected
Input: 12	In B1
Input: 13	In A1

Table A.9: Pinout For 7418

A.10 7419: HEX INVERTER (SCHMITT-TRIGGER)

This device contains six independent inverters. Schmitt-triggers are a special type of device that are used to filter out spurious noise on a circuit. They are designed to change from low-to-high or high-to-low only when the input voltage reaches a preset level but not if the voltage randomly fluctuates without crossing the set-points. This device is essentially the same as the 7414. Figure A.11 is a logic diagram of one of the six circuits.

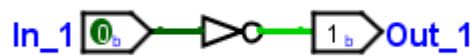


Figure A.11: 7419: Single Inverter Circuit

The 7419 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.10.

Logisim Label	Function
Input: 1	In 1
Output: 2	Out 1
Input: 3	In 2
Output: 4	Out 2
Input: 5	In 3
Output: 6	Out 3
Output: 8	Out 4
Input: 9	In 4
Output: 10	Out 5
Input: 11	In 5
Output: 12	Out 6
Input: 13	In 6

Table A.10: Pinout For 7419

A.11 7420: DUAL 4-INPUT NAND GATE

This device contains two independent 4-input NAND gates. Figure [A.12](#) is a logic diagram of one of the two circuits.

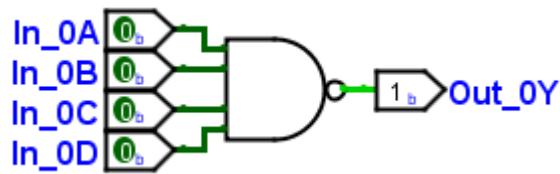


Figure A.12: 7420: Single 4-Input NAND Gate Circuit

The 7420 device in *Logisim-Evolution* uses the wiring connections indicated in Table [A.11](#).

Logisim Label	Function
Input: 1	In A0
Input: 2	In B0
Pin 3 NC	Not Connected
Input: 4	In C0
Input: 5	In D0
Output: 6	Out Y0
Output: 8	Out Y1
Input: 9	In D1
Input: 10	In C1
Pin 11 NC	Not Connected
Input: 12	In B1
Input: 13	In A1

Table A.11: Pinout For 7420

A.12 7421: DUAL 4-INPUT AND GATE

This device contains two independent 4-input AND gates. Figure A.13 is a logic diagram of one of the two circuits.

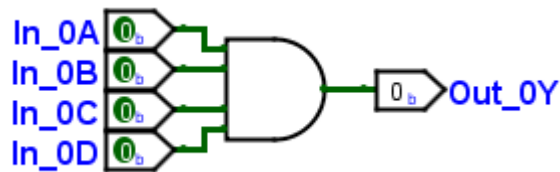


Figure A.13: 7421: Single 4-Input AND Gate Circuit

The 7421 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.12.

Logisim Label	Function
Input: 1	In A0
Input: 2	In B0
Pin 3 NC	Not Connected
Input: 4	In C0
Input: 5	In D0
Output: 6	Out Y0
Output: 8	Out Y1
Input: 9	In D1
Input: 10	In C1
Pin 11 NC	Not Connected
Input: 12	In B1
Input: 13	In A1

Table A.12: Pinout For 7421

A.13 7424: QUAD 2-INPUT NAND GATE (SCHMITT-TRIGGER)

This device contains four independent 2-input NAND gates. Schmitt-triggers are a special type of device that are used to filter out spurious noise on a circuit. They are designed to change from low-to-high or high-to-low only when the input voltage reaches a preset level but not if the voltage randomly fluctuates without crossing the set-points. This device is essentially the same as the 7400. Figure A.14 is a logic diagram of one of the four circuits.

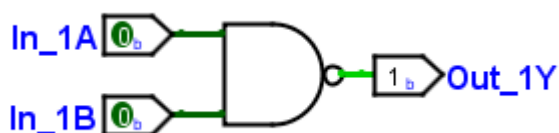


Figure A.14: 7424: Single NAND Gate Circuit

The 7424 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.13.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.13: Pinout For 7424

A.14 7427: TRIPLE 3-INPUT NOR GATE

This device contains three independent 3-input NOR gates. Figure A.15 is a logic diagram of one of the three circuits.

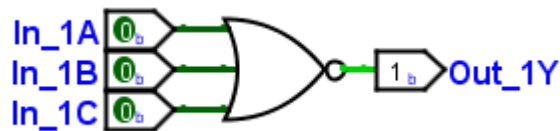


Figure A.15: 7411: Single 3-Input NOR Gate Circuit

The 7427 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.14.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Input: 3	In 2A
Input: 4	In 2B
Input: 5	In 2C
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Input: 11	In 3C
Output: 12	Out 1Y
Input: 13	In 1C

Table A.14: Pinout For 7427

A.15 7430: SINGLE 8-INPUT NAND GATE

This device contains a single 8-input NAND gate. The logic for this gate is $Y = \overline{A \cdot B \cdot C \cdot D \cdot E \cdot F \cdot G \cdot H}$. Figure A.16 is a logic diagram of the circuit.

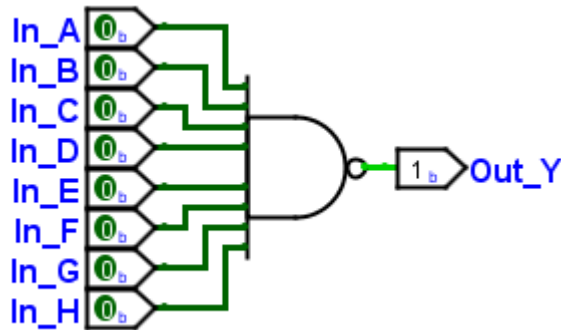


Figure A.16: 7430: Single 8-Input NAND Gate

The 7430 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.15.

Logisim Label	Function
Input: 1	In A
Input: 2	In B
Input: 3	In C
Input: 4	In D
Input: 5	In E
Input: 6	In F
Output: 8	Out Y
Pin 9: NC	Not Connected
Pin 10: NC	Not Connected
Input: 11	In G
Input: 12	In H
Pin 13: NC	Not Connected

Table A.15: Pinout For 7430

A.16 7432: QUAD 2-INPUT OR GATE

This device contains four independent 2-input OR gates. Figure A.17 is a logic diagram of one of the four circuits.

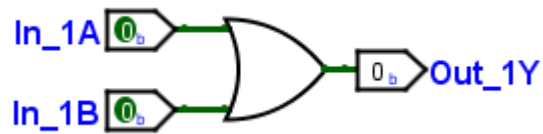


Figure A.17: 7432: Single OR Gate Circuit

The 7432 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.16.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.16: Pinout For 7432

A.17 7436: QUAD 2-INPUT NOR GATE

This device contains four independent 2-input NOR gates. This device is essentially the same as the 7402. Figure A.18 is a logic diagram of one of the four circuits.

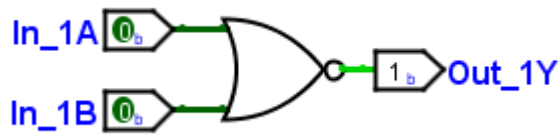


Figure A.18: 7436: Single NOR Gate Circuit

The 7436 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.17.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.17: Pinout For 7436

A.18 7442: BCD TO DECIMAL DECODER

This device takes a BCD input and deactivates a single line corresponding to the input number. It is often called a “One-Of-Ten” decoder. As an example, if 0111_{BCD} is input then line 7-of-10 will go low while all other outputs will remain high. Figure A.19 illustrates a 7442 IC in a very simple circuit.

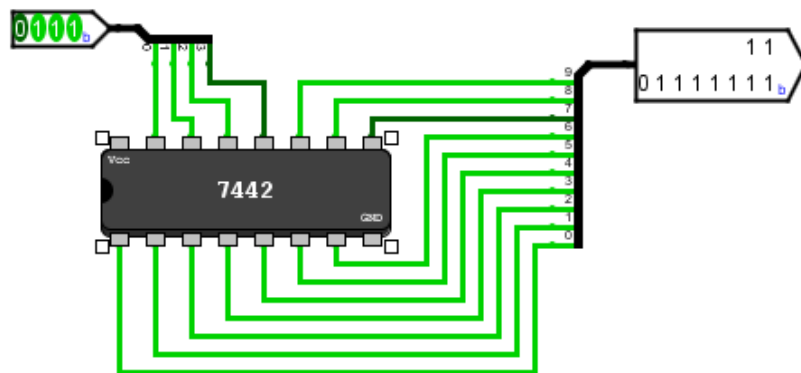


Figure A.19: 7442: BCD to Decimal Decoder

Table A.18 is the truth table for this device. Any BCD input greater than 1001 is ignored and all outputs will be high for those inputs.

Inputs				Output									
A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

Table A.18: Truth Table For The 7442 Circuit

The 7442 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.19.

Logisim Label	Function
Output 1: O0	Out 0
Output 2: O1	Out 1
Output 3: O2	Out 2
Output 4: O3	Out 3
Output 5: O4	Out 4
Output 6: O5	Out 5
Output 7: O6	Out 6
Output 8: O7	Out 7
Output 10: O8	Out 8
Output 11: O9	Out 9
Input 12: D	In D
Input 13: C	In C
Input 14: B	In B
Input 15: A	In A

Table A.19: Pinout For 7442

A.19 7443: EXCESS-3 TO DECIMAL DECODER

This device takes an Excess-3 input and deactivates a single line corresponding to the input number. It is often called a “One-Of-Ten”

decoder. As an example, if 0011_{Ex3} is input then line 0-of-10 will go low while all other outputs will remain high. This is wired in exactly the same way as the 7442 IC illustrated in Figure A.19.

Table A.20 is the truth table for this device. Any input numbers other than those found in the truth table are ignored and all outputs will be high for those inputs.

Inputs				Output									
A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	1	1	1	1	1	1	1	1	1
0	1	0	0	1	0	1	1	1	1	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	1	1	1	1	0

Table A.20: Truth Table For The 7443 Circuit

The 7443 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.21.

Logisim Label	Function
Output 1: O0	Out 0
Output 2: O1	Out 1
Output 3: O2	Out 2
Output 4: O3	Out 3
Output 5: O4	Out 4
Output 6: O5	Out 5
Output 7: O6	Out 6
Output 8: O7	Out 7
Output 10: O8	Out 8
Output 11: O9	Out 9
Input 12: D	In D
Input 13: C	In C
Input 14: B	In B
Input 15: A	In A

Table A.21: Pinout For 7443

A.20 7444: GRAY TO DECIMAL DECODER

This device takes a Gray Excess Code, which is a combination of Gray and Excess-3 Codes, input and deactivates a single line corresponding to the input number. It is often called a “One-Of-Ten” decoder. As an example, if 1100_{GrayEx3} is input then line 5-of-10 will go low while all other outputs will remain high. This is wired in exactly the same way as the 7442 IC illustrated in Figure A.19.

Table A.22 is the truth table for this device. Any input numbers other than those found in the truth table are ignored and all outputs will be high for those inputs.

Inputs				Output									
A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	1	1	1	1	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1	1	1
0	1	0	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	0	1	1	1	1
1	1	0	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	1	1	1	1	1	1	0

Table A.22: Truth Table For The 7444 Circuit

The 7443 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.23.

Logisim Label	Function
Output 1: O0	Out 0
Output 2: O1	Out 1
Output 3: O2	Out 2
Output 4: O3	Out 3
Output 5: O4	Out 4
Output 6: O5	Out 5
Output 7: O6	Out 6
Output 8: O7	Out 7
Output 10: O8	Out 8
Output 11: O9	Out 9
Input 12: D	In D
Input 13: C	In C
Input 14: B	In B
Input 15: A	In A

Table A.23: Pinout For 7444

A.21 7447: BCD TO 7-SEGMENT DECODER

This device takes a BCD Code input and activates a combination of outputs such that a 7-segment display will correctly indicate the input number. Figure A.20 illustrates a 7447 IC in a very simple circuit.

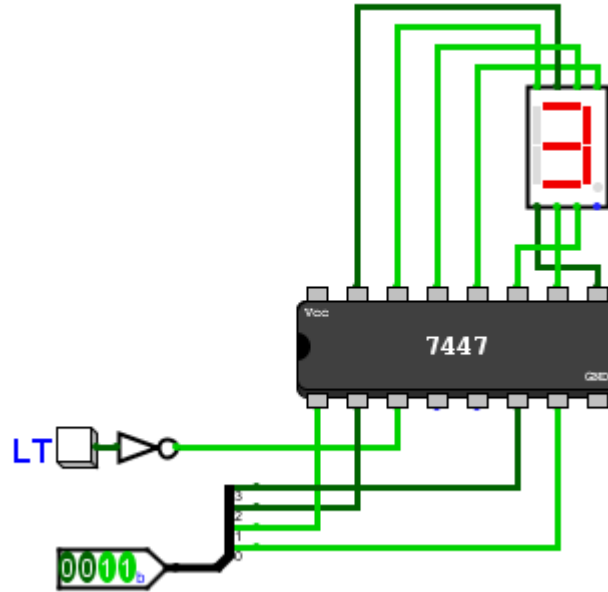


Figure A.20: 7447: BCD to 7-Segment Decoder

Table A.24 is the truth table for this device.

Inputs				Output						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	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
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1

Table A.24: Truth Table For The 7447 Circuit

The 7447 device in *Logisim-Evolution* uses the wiring connections indicated in Table [A.25](#).

Logisim Label	Function
Input 1: B	B
Input 2: C	C
Input 3: LT	LT
Input 4: BI	BI
Input 5: RBI	RBI
Input 6: D	D
Input 7: A	A
Output 8: e	e
Output 10: d	d
Output 11: c	c
Output 12: b	b
Output 13: a	a
Output 14: g	g
Output 15: f	f

Table A.25: Pinout For 7447

A.22 7451: DUAL AND-OR-INVERT GATE

This device contains two independent AND-OR-INVERT gates. Figure A.21 is a logic diagram of one of the two circuits.

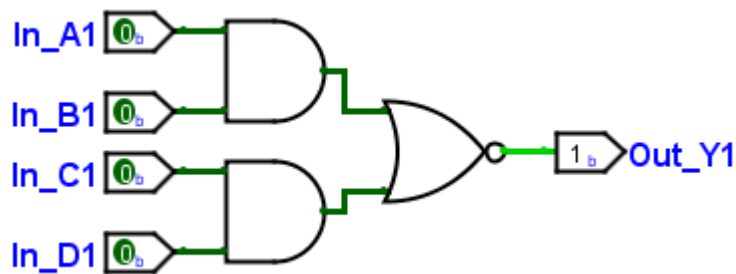


Figure A.21: 7451: Single AND-OR-INVERT Gate Circuit

The 7451 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.26.

Logisim Label	Function
Input 1: A1	In A1
Input 2: A2	In A2
Input 3: B2	In B2
Input 4: C2	In C2
Input 5: D2	In D2
Output 6: Y2	Out Y2
Output 8: Y1	Out Y1
Input 9: C1	In C1
Input 10: D1	In D1
Pin 11: NC	Not Connected
Pin 12: NC	Not Connected
Input 13: B1	In B1

Table A.26: Pinout For 7451

A.23 7454: FOUR WIDE AND-OR-INVERT GATE

This device contains a single four-wide AND-OR-INVERT gate. Figure A.22 is a logic diagram of the circuit.

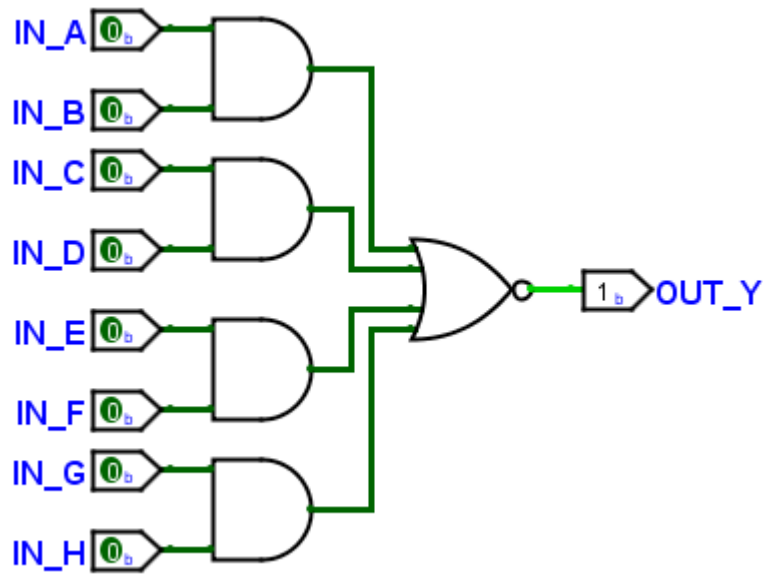


Figure A.22: 7454: Four Wide AND-OR-INVERT Gate Circuit

The 7454 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.27.

Logisim Label	Function
Input 1: A	In A
Input 2: C	In C
Input 3: D	In D
Input 4: E	In E
Input 5: F	In F
Pin 6: NC	Not Connected
Output 8: Y	Out Y
Input 9: G	In G
Input 10: H	In H
Pin 11: NC	Not Connected
Pin 12: NC	Not Connected
Input 13: B	In B

Table A.27: Pinout For 7454

A.24 7458: DUAL AND-OR GATE

This device contains a two AND-OR gates. One has three-input AND gates and the other has two-input AND gates. Figure A.23 is a logic diagram of the circuit.

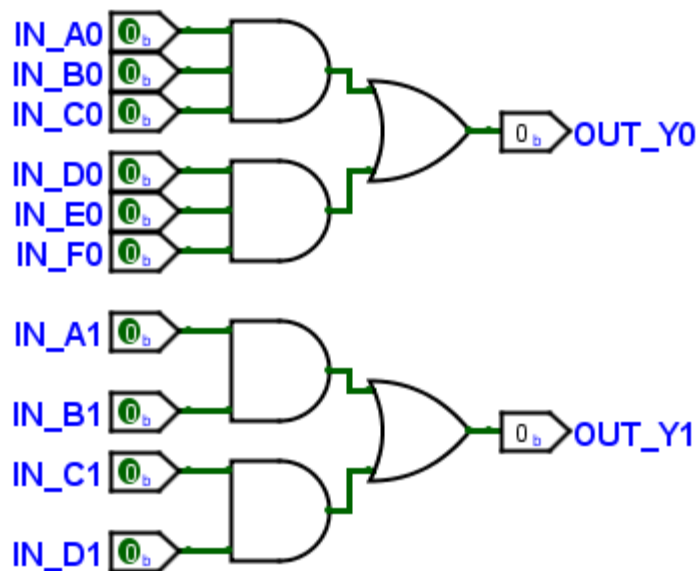


Figure A.23: 7458: Dual AND-OR Gate Circuit

The 7458 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.28.

Logisim Label	Function
Input 1: A0	In A0
Input 2: A1	In A1
Input 3: B1	In B1
Input 4: C1	In C1
Input 5: D1	In D1
Output 6: Y1	Out Y1
Output 8: Y0	Out Y0
Input 9: D0	In D0
Input 10: E0	In E0
Input 11: F0	In F0
Input 12: B0	In B0
Input 13: C0	In C0

Table A.28: Pinout For 7458

A.25 7464: 4-2-3-2 AND-OR-INVERT GATE

This device contains four AND gates of different input sizes that feed a NOR gate. Figure A.24 is a logic diagram of the circuit.

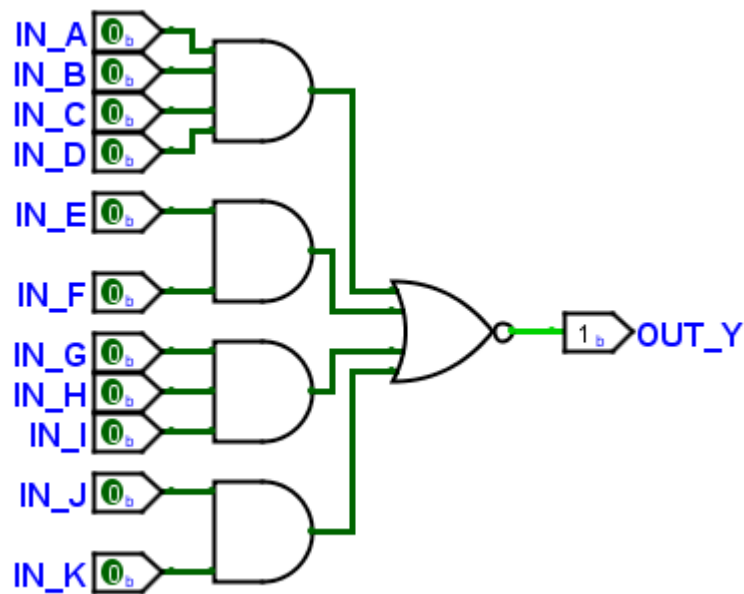


Figure A.24: 7464: 4-2-3-2 AND-OR-INVERT Gate Circuit

The 7464 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.29.

Logisim Label	Function
Input 1: A	In A
Input 2: E	In E
Input 3: F	In F
Input 4: G	In G
Input 5: H	In H
Input 6: I	In I
Output 8: Y	Out Y
Input 9: J	In J
Input 10: K	In K
Input 11: B	In B
Input 12: C	In C
Input 13: D	In D

Table A.29: Pinout For 7464

A.26 7474: DUAL D-FLIPFLOPS WITH PRESET AND CLEAR

This device contains two D-Flipflops, each with its own preset and clear. The 7474 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.30.

Logisim Label	Function
Input 1: nCLR1	On low, clear FF1
Input 2: D1	FF1 data input
Input 3: CLK1	FF1 clock
Input 4: nPRE1	On low, set FF1
Output 5: Q1	FF1 Q-out
Output 6: nQ1	FF1 Q-not-out
Output 8: nQ2	FF2 Q-not-out
Output 9: Q2	FF2 Q-out
Input 10: nPRE2	On low, set FF2
Input 11: CLK2	FF2 clock
Input 12: D2	FF2 data input
Input 13: nCLR2	On low, clear FF2

Table A.30: Pinout For 7474

A.27 7485: 4-BIT MAGNITUDE COMPARATOR

This device compares two 4-bit numbers and outputs one of three values: $A > B$, $A = B$, or $A < B$. It is also designed to be cascaded by including an input port for each of the three values. The 7485 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.31.

Logisim Label	Function
Input 1: B3	Bit B3
Input 2: A<B	Value from prior stage
Input 3: A=B	Value from prior stage
Input 4: A>B	Value from prior stage
Output 5: A>B	High if $A > B$
Output 6: A=B	High if $A = B$
Output 7: A<B	High if $A < B$
Input 9: B0	Bit B0
Input 10: A0	Bit A0
Input 11: B1	Bit B1
Input 12: A1	Bit A1
Input 13: A2	Bit A2
Input 14: B2	Bit B2
Input 15: A3	Bit A3

Table A.31: Pinout For 7485

A.28 7486: QUAD 2-INPUT XOR GATE

This device contains four independent 2-input XOR gates. Figure A.25 is a logic diagram of one of the four circuits.

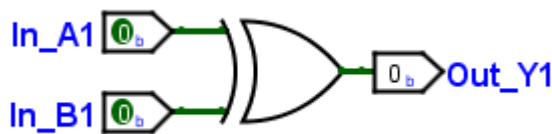


Figure A.25: 7486: Single XOR Gate Circuit

The 7486 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.32.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.32: Pinout For 7486

A.29 74125: QUAD BUS BUFFER, 3-STATE GATE

This device contains four independent buffers. When each is enabled with a low on the enable line then the input is passed to the output, when not enabled then the output floats. Figure A.26 is a logic diagram of one of the four circuits.

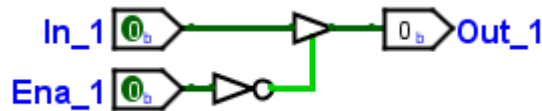


Figure A.26: 74125: Single Buffer Circuit

The 74125 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.33.

Logisim Label	Function
Input: 1	nEna 1
Input: 2	In 1
Output: 3	Out 1
Input: 4	nEna 2
Input: 5	In 2
Output: 6	Out 2
Output: 8	Out 3
Input: 9	In 3
Input: 10	nEna 3
Output: 11	Out 4
Input: 12	In 4
Input: 13	nEna 4

Table A.33: Pinout For 74125

A.30 74165: 8-BIT PARALLEL-TO-SERIAL SHIFT REGISTER

This device can accept data in either parallel or serial form and shift it out in serial form. The 74165 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.34.

Logisim Label	Function
Input 1: Shift/Load	Load when low, shift when high
Input 2: Clock	Clock
Input 3: P4	Input bit 4
Input 4: P5	Input bit 5
Input 5: P6	Input bit 6
Input 6: P7	Input bit 7
Output 7: Q7n	Complement of serial out
Output 9: Q7	Serial out
Input 10: Serial Input	Serial data in
Input 11: P0	Input bit 0
Input 12: P1	Input bit 1
Input 13: P2	Input bit 2
Input 14: P3	Input bit 3
Input 15: Clock Inhibit	Clock inhibit

Table A.34: Pinout For 74165

A.31 74175: QUAD D-FLIPFLOPS WITH SYNC RESET

This device contains four D-Flipflops with a single clock and master reset. The 74175 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.35.

Logisim Label	Function
Input 1: nCLR	On low, clear all FF
Output 2: Q1	FF1 Q-out
Output 3: nQ1	FF1 Q-not-out
Input 4: D1	FF1 data input
Input 5: D2	FF2 data input
Output 6: nQ2	FF2 Q-not-out
Output 7: Q2	FF2 Q-out
Input 9: CLK	Clock for all FF
Output 10: Q3	FF3 Q-out
Output 11: nQ3	FF3 Q-not-out
Input 12: D3	FF3 data input
Input 13: D4	FF4 data input
Output 14: nQ4	FF4 Q-not-out
Output 15: Q4	FF4 Q-out

Table A.35: Pinout For 74175

A.32 74266: QUAD 2-INPUT XNOR GATE

This device contains four independent 2-input XNOR gates. Figure A.27 is a logic diagram of one of the four circuits.

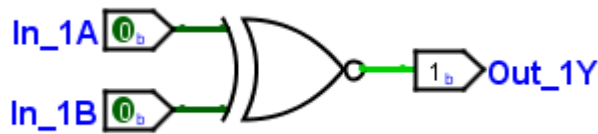


Figure A.27: 74266: Single XNOR Gate Circuit

The 74266 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.36.

Logisim Label	Function
Input: 1	In 1A
Input: 2	In 1B
Output: 3	Out 1Y
Input: 4	In 2A
Input: 5	In 2B
Output: 6	Out 2Y
Output: 8	Out 3Y
Input: 9	In 3A
Input: 10	In 3B
Output: 11	Out 4Y
Input: 12	In 4A
Input: 13	In 4B

Table A.36: Pinout For 74266

A.33 74273: OCTAL D-FLIPFLOP WITH CLEAR

This device contains a single 8-bit D-Flipflop with a single clock and master clear. The 74273 device in *Logisim-Evolution* uses the wiring connections indicated in Table [A.37](#).

Logisim Label	Function
Input 1: nCLR	On low, clear the FF
Output 2: Q1	data bit 1 output
Input 3: D1	data bit 1 input
Input 4: D2	data bit 2 input
Output 5: Q2	data bit 2 output
Output 6: Q3	data bit 3 output
Input 7: D3	data bit 3 input
Input 8: D4	data bit 4 input
Output 9: Q4	data bit 4 output
Input 11: CLK	Clock
Output 12: Q5	data bit 5 output
Input 13: D5	data bit 5 input
Input 14: D6	data bit 6 input
Output 15: Q6	data bit 6 output
Output 16: Q7	data bit 7 output
Input 17: D7	data bit 7 input
Input 18: D8	data bit 8 input
Output 19: Q8	data bit 8 output

Table A.37: Pinout For 74273

A.34 74283: 4-BIT BINARY FULL ADDER

This device contains a 4-bit adder with carry-in and carry-out bits. The 74283 device in *Logisim-Evolution* uses the wiring connections indicated in Table [A.38](#).

Logisim Label	Function
Output 1: $\sum 2$	Sum, bit 2
Input 2: B2	Operand B, bit 2
Input 3: A2	Operand A, bit 2
Output 4: $\sum 1$	Sum, bit 1
Input 5: A1	Operand A, bit 1
Input 6: B1	Operand B, bit 1
Input 7: CIN	Carry in bit
Output 9: C4	Carry out bit
Output 10: $\sum 4$	Sum, bit 4
Input 11: B4	Operand B, bit 4
Input 12: A4	Operand A, bit 4
Output 13: $\sum 3$	Sum, bit 3
Input 14: A3	Operand A, bit 3
Input 15: B3	Operand B, bit 3

Table A.38: Pinout For 74283

A.35 74377: OCTAL D-FLIPFLOP WITH ENABLE

This device contains a single 8-bit D-Flipflop with a single clock and enable. The 74377 device in *Logisim-Evolution* uses the wiring connections indicated in Table A.39.

Logisim Label	Function
Input 1: nCLKen	On low, enable the clock
Output 2: Q1	data bit 1 output
Input 3: D1	data bit 1 input
Input 4: D2	data bit 2 input
Output 5: Q2	data bit 2 output
Output 6: Q3	data bit 3 output
Input 7: D3	data bit 3 input
Input 8: D4	data bit 4 input
Output 9: Q4	data bit 4 output
Input 11: CLK	Clock
Output 12: Q5	data bit 5 output
Input 13: D5	data bit 5 input
Input 14: D6	data bit 6 input
Output 15: Q6	data bit 6 output
Output 16: Q7	data bit 7 output
Input 17: D7	data bit 7 input
Input 18: D8	data bit 8 input
Output 19: Q8	data bit 8 output

Table A.39: Pinout For 74377

COLOPHON

This book was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and \LyX :

<https://bitbucket.org/amiede/classicthesis/>

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

<http://postcards.miede.de/>

Final Version as of August 17, 2019 (`classicthesis` Edition 4.0).

Hermann Zapf's *Palatino* and *Euler* type faces (Type 1 PostScript fonts *URW Palladio L* and *FPL*) are used. The "typewriter" text is typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera". (Type 1 PostScript fonts were made available by Malte Rosenau and Ulrich Dirr.)

