MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 23 02
F: +90 312 210 23 04
ee@metu.edu.tr
www.eee.metu.edu.tr

EXPERIMENT 5. ELEMENTARY GATE NETWORKS

I. Introduction

1) Objectives

In this experiment, you will get familiar with some of commonly used elementary gate networks. In particular, multiplexer, comparator and some code converters will be investigated and some use-cases will be studied. Throughout this experiment, logic gate networks will be implemented by using both discrete components and Quartus II development environment.

2) Background

2.a) Multiplexers

A multiplexer is a combinational circuit controlled by selectors, which selects and directs one of several input lines to its output. Conventionally and ideally, a multiplexer is a combination of m groups of n input signals per group and k select lines, where $n=2^k$. However, there might be less inputs per groups than 2^k. A generic block diagram of an m-input n-bit multiplexer is given in Fig. 1.

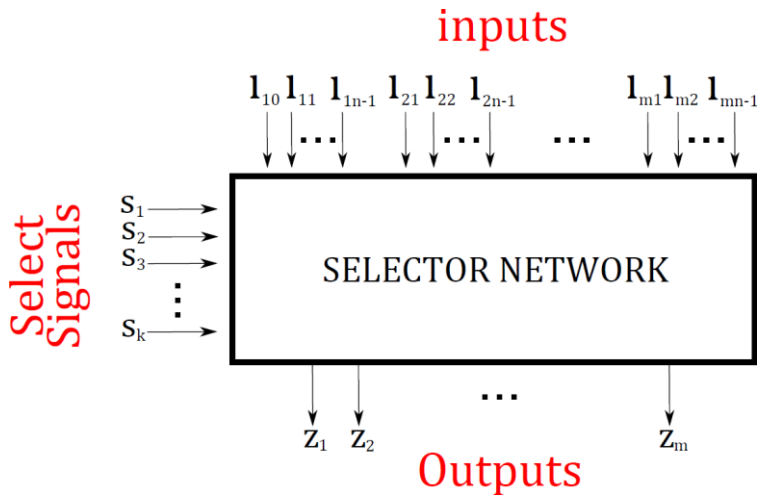


Fig. 1: Block diagram of an m-input n-bit multiplexer

Based on select inputs $S_0\ S_1 \ldots S_{k-1}$, output pins are controlled, e.g., if all select bits are configured as "0", output becomes $I_{01}\ I_{01} \ldots I_{(m-1)1}$. As an example consider the 3-input 2x1multiplexer given in Fig. 2. In this configuration, $\boldsymbol{I_a(I_{a1}I_{a2})}$, $\boldsymbol{I_b(I_{b1}I_{b2})}$, $\boldsymbol{I_c(I_{c1}I_{c2})}$ are the input groups, $\boldsymbol{Z_1, Z_2, Z_3}$ are outputs. Depending on the state of select input, $\boldsymbol{S_1}$, output $\boldsymbol{Z_i}$, becomes equal to either $\boldsymbol{I_{i1}}$or $\boldsymbol{I_{i2}}$, where $\boldsymbol{i} \in \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$. Additionally, enable pin $\boldsymbol{E}$ is an active-low control signal, where output $\boldsymbol{Z_1\ Z_2\ Z_3 = 0}$ when $\boldsymbol{E}$**=1**.

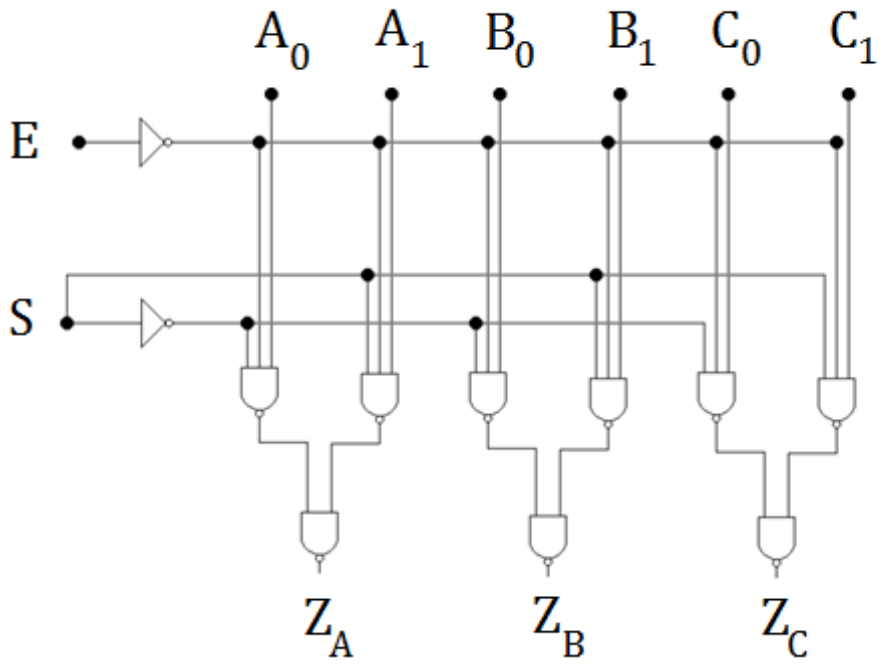


Fig. 2: 3-input 2-to-1 (2x1) multiplexer

2.b) Comparators

In digital design, comparators are used for magnitude comparison, where two *n*-bit numbers are inputs of the circuit. Comparators determine whether a given *n*-bit number is equal to, greater or less than the second number. Conventionally, a comparator circuit has 3 outputs corresponding to the equal, less and greater cases. As there are 3 possible output cases, an alternative approach could be the indication of these 3 cases by 2-bits. In Table-1, truth table of a 2-bit comparator with 2 outputs is given. In this truth table, two 2-bit numbers, i.e., A and B, are compared by means of their magnitude and the output is 10, 01 and 11 when A>B, A<B and A=B, respectively.

A		B		Z	
A_1	A_0	B_1	B_0	Z_1	Z_0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	1

Table-1: Truth table of an alternative 2-bit magnitude comparator

2.c) Code Converters

Although binary codes are widely used, in some applications other coding systems are preferred. In rotary encoders, which track angular position changes, gray coding is in use. The gray code has the property that a codeword in the code sequence differ by only one bit compared to its predecessor and successor codewords. Table-2 gives the truth table of a 4-bit binary to gray code converter.

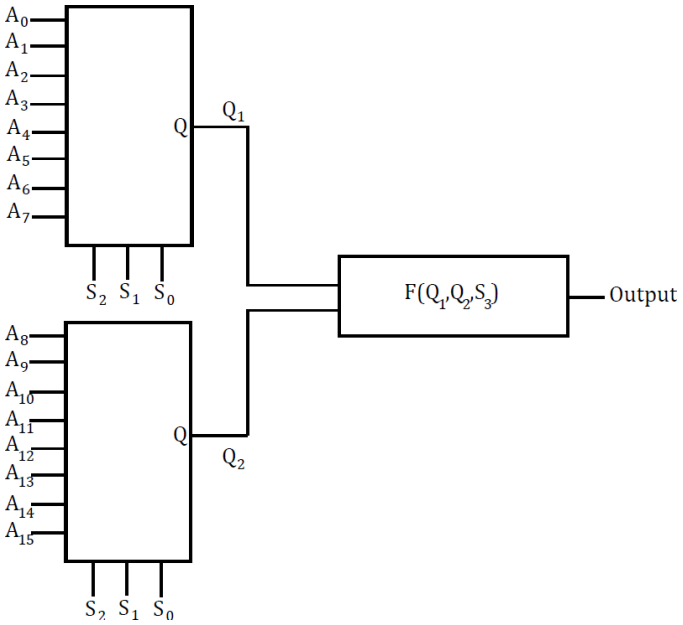
	Binary				Gray			
Decimal	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Table-2: Truth table of a 4-bit binary to gray code converter

II. Preliminary Work

1) Read section 4.8 - 4.11 of your course book, "Digital Design" (M. Morris Mano, Prentice Hall) and revise manuals of experiments 1, 3 and 4 to remember FPGA programming and hierarchical design processes.

2) Consider the logic circuit given below. It is known that this circuit is a 16-to-1 multiplexer. What should $F(Q_1, Q_2, S_3)$ block be?



Inputs: $A_0A_1 \ldots A_{15}$
Select Signals: $S_3S_2S_1S_0$

3) Design a 16-to-1 multiplexer using 74151 ICs and NAND gates. Refer to Appendix A, for pin diagram and truth table of 74151.

4) Keeping the hierarchical design in mind, design another 16-to-1 multiplexer using only 4-input, 3-input, 2-input NAND gates and inverters. After designing sub-blocks on gate level, you can use block demonstrations.

5) Design a comparator circuit to compare two 4-bit numbers. As shown in Table-1, your design should have 2 outputs, which takes the values 10, 01 and 11 for A>B, A<B and A=B cases, respectively. This design will be implemented in Quartus II development environment and you are not allowed to use any complex block. You are allowed to use only NAND, NOR, XOR and INVERTER gates in your design.

6) Using the truth table given in Table-2, design a logic circuit, which converts a 4-bit binary code to gray when select input is 1, whereas it converts gray code to binary when the select input is 0.

III. Experimental Work

1) Construct the 16-bit multiplexer that you have designed in preliminary work Part-3 on the protoboard. Afterwards, test and verify that your design works as expected.

2) Draw the circuit that you have designed in preliminary work Part-4 using Quartus II development tools. You are expected to build a hierarchical design, where final 16-to-1 multiplexer is formed by 4-to-1 multiplexer blocks. First simulate the circuit and then implement it on FPGA board. It would be wise to use buses for select inputs. Make sure both your simulation and implementation results are consistent with the expectations. During development and implementation process if you need help, you can refer to manuals of experiments 1, 3 and 4.

Important Note 1: A 16-to-1 MUX has 4 select and 16 input pins. De1-SoC board, on the other hand, contains only 10 slide switches that we use as input elements. Fortunately, we have also two 2x20 General Purpose Input/Output (GPIO) pins as shown in Fig. 3. Each GPIO port contains 36 user pins directly connected to the FPGA, DC 5V, DC 3.3V and two GND pins, as depicted in Fig. 4.

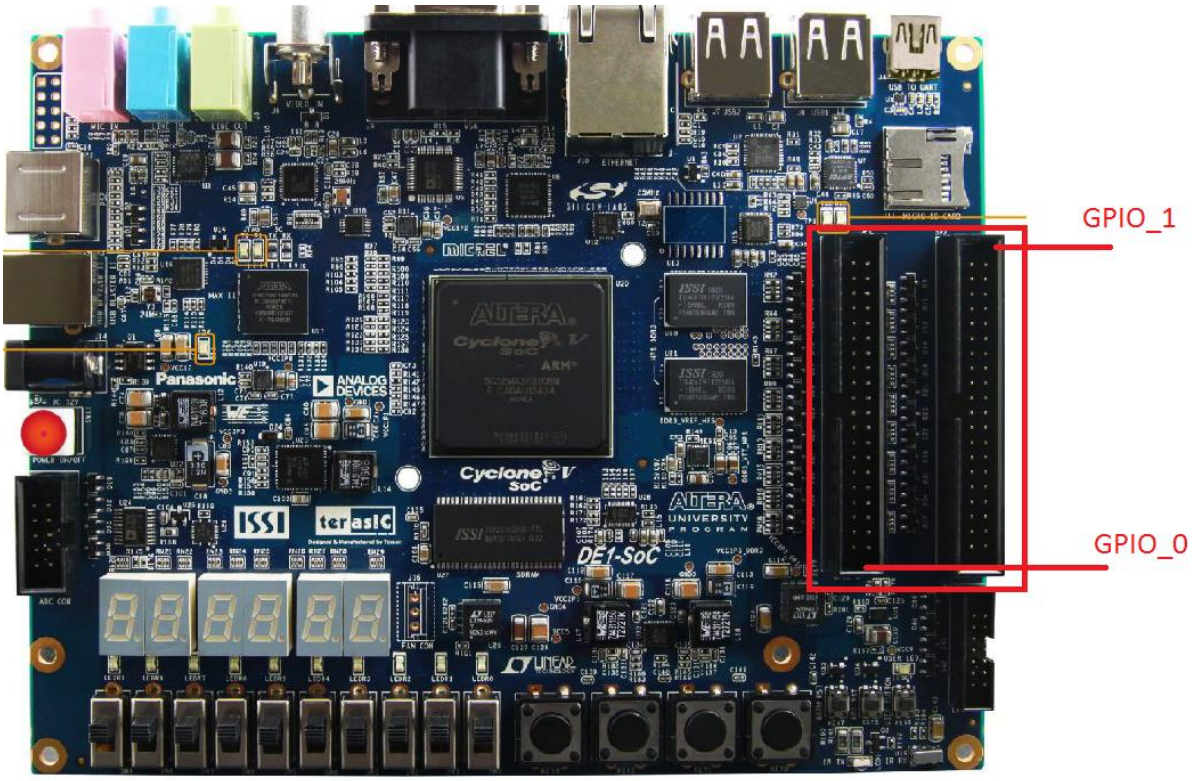


Fig.3 : GPIO Pins

In this part of the experiment, we will use 10 of these user pins to provide remaining 10 inputs. As the I/O standard of the user pins is 3.3V, VCC3P3_GPIO and

GND pins will also be used. To this end, jumper wires will be connected to the corresponding pins and signals will be carried on a protoboard. First of all, wire the female end of a jumper to VCC3P3_GPIO pin and connect the male end to signal port of your protoboard. Next, do the same for GND pin and connect male end to the GND port of the protoboard. After these steps, connections should be made as illustrated in Fig.5. Finally, connect female ends of 10 jumper wires to the user pins that you select and leave the male ends unconnected. You will apply 3.3V or GND from these male ends to verify your design. Based on the selection of which 10 pins among 36 user pins are used, proper pin assignments must be made according to the FPGA pin numbers of GPIO banks listed in Appendix B.

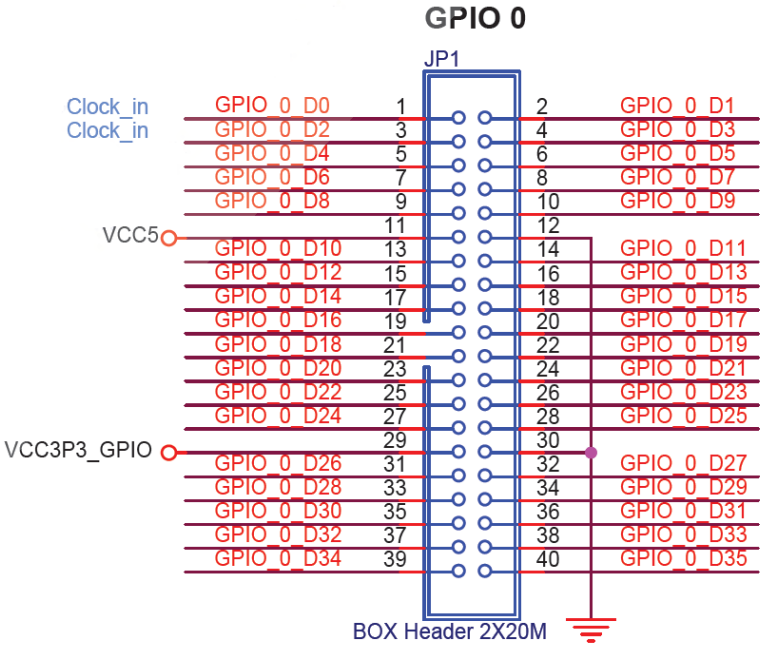


Fig. 4: Pin diagram of GPIO 0

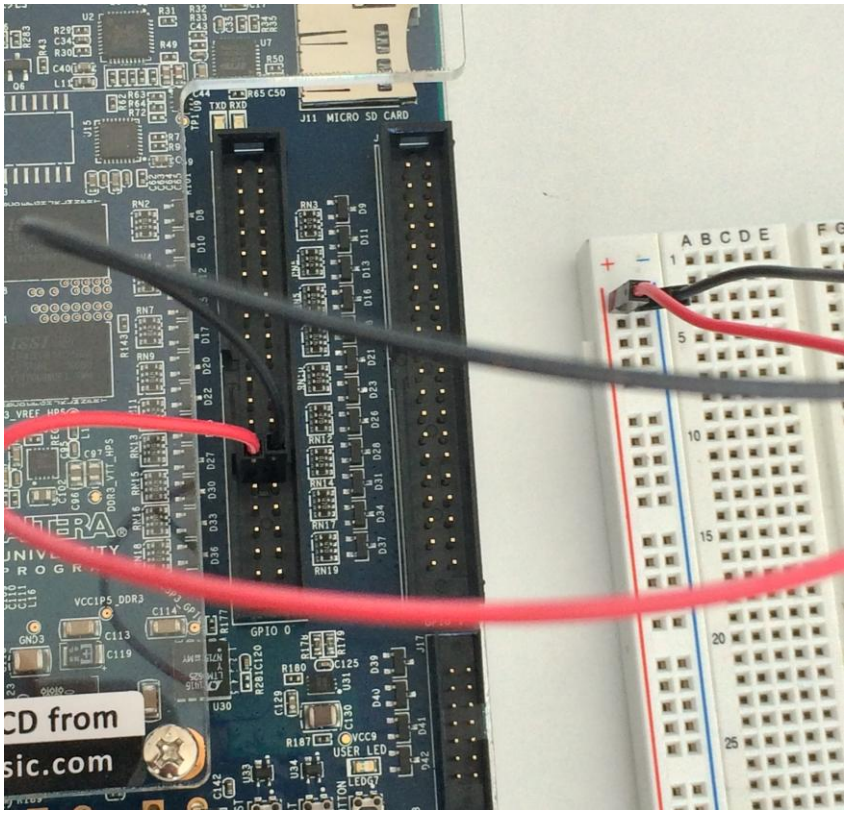


Fig. 5: Wiring signal and gnd pins to a protobard

Important Note 2: In the previous experiment during hierarchical design steps, you have created a new project for each low-level design block. It is also possible to build symbol files within a single project. To do so, first create a new project and a new schematic file. Make sure "Add file to current project" checkbox is selected. Name the schematic same as the project name. Next, create another project file and make sure it is also added to the current project. Draw your low-level design and save it. Don't forget to give a proper name, i.e., a convenient name describing functionality of the sub-block which is different than the project name. Finally, create a new symbol from the schematic **(File > Create/Update > Create Symbol Files for Current File)**. Now, you can use the symbol block within the top-level schematic file of your project, i.e. *project_name.bdf*.

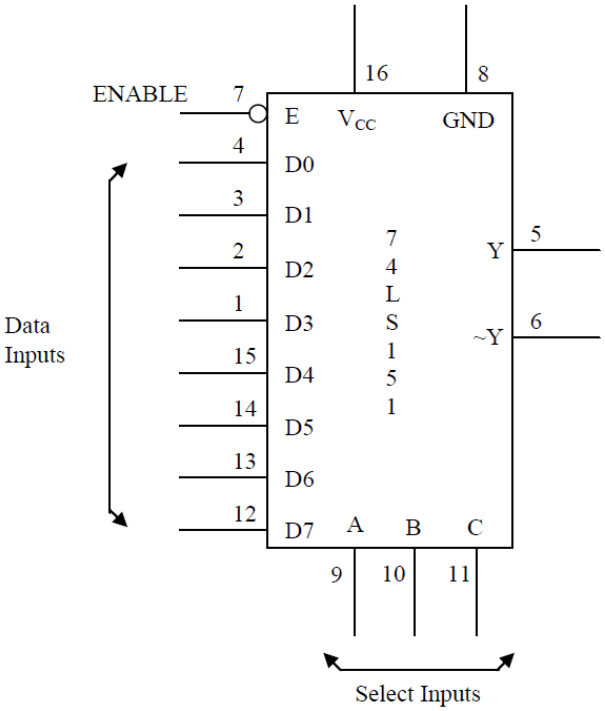
3) Draw the comparator circuit that you have designed in preliminary work Part-5 using Quartus II schematic editor. Verify your design using simulation and then program the FPGA and perform final tests on hardware. In this step, you are required to use buses for inputs representing each 4-bit number.

4) Draw the binary to gray and gray to binary code converter circuits that you have designed in preliminary work Part-6 using Quartus II schematic editor. Verify your design using simulation and then program the FPGA and perform final tests on hardware.

Appendix A: Part List and Pin Diagrams

74151IC: 8-to-1 MUX with active low enable
7400 IC: 2 input NAND gates
7404 IC: Inverter gates
7410 IC: 3 input NAND gates
7486 IC: 2 input XOR gates
7420 IC: 4 input NAND gates

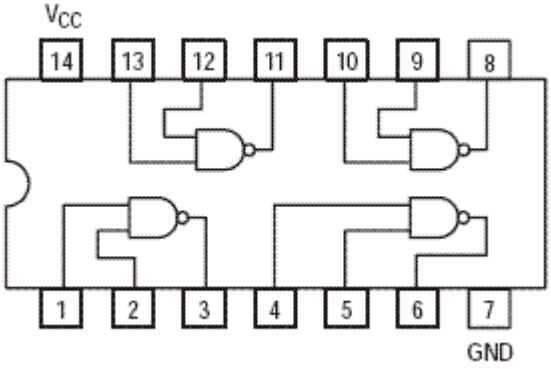
74151:



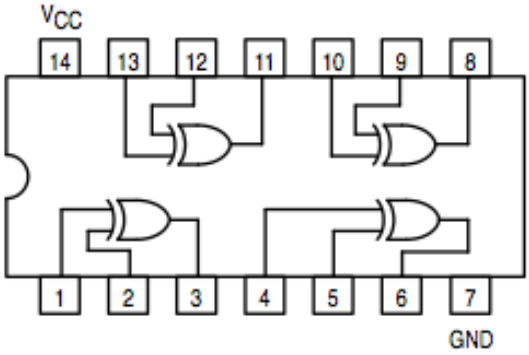
E : Active-low enable pin
D0-D7: Input pins
Y : Output
~Y : Inverse of output

E	A	B	C	Y
1	X	X	X	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

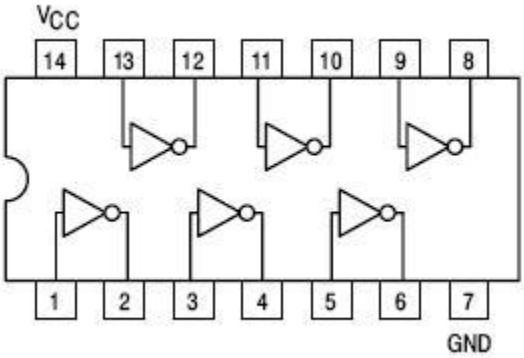
7400:



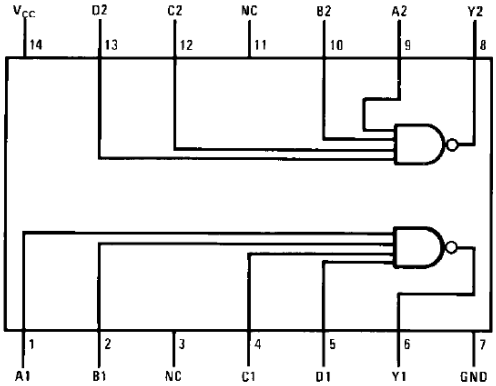
7486:



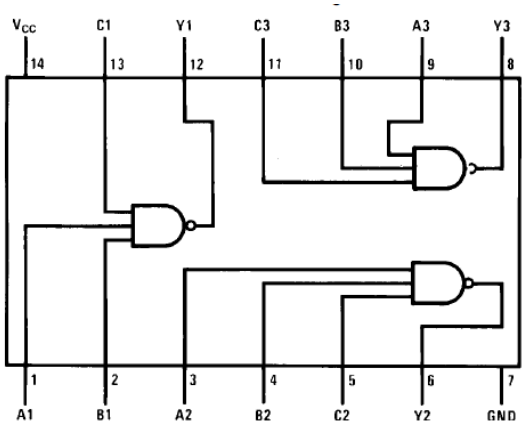
7404:



7420:



7410:



Appendix B: FPGA switch, button, GPIO and LED description table

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB12	Slide Switch[0]	3.3V
SW[1]	PIN_AC12	Slide Switch[1]	3.3V
SW[2]	PIN_AF9	Slide Switch[2]	3.3V
SW[3]	PIN_AF10	Slide Switch[3]	3.3V
SW[4]	PIN_AD11	Slide Switch[4]	3.3V
SW[5]	PIN_AD12	Slide Switch[5]	3.3V
SW[6]	PIN_AE11	Slide Switch[6]	3.3V
SW[7]	PIN_AC9	Slide Switch[7]	3.3V
SW[8]	PIN_AD10	Slide Switch[8]	3.3V
SW[9]	PIN_AE12	Slide Switch[9]	3.3V

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
KEY[0]	PIN_AA14	Push-button[0]	3.3V
KEY[1]	PIN_AA15	Push-button[1]	3.3V
KEY[2]	PIN_W15	Push-button[2]	3.3V
KEY[3]	PIN_Y16	Push-button[3]	3.3V

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LEDR[0]	PIN_V16	LED [0]	3.3V
LEDR[1]	PIN_W16	LED [1]	3.3V
LEDR[2]	PIN_V17	LED [2]	3.3V
LEDR[3]	PIN_V18	LED [3]	3.3V
LEDR[4]	PIN_W17	LED [4]	3.3V
LEDR[5]	PIN_W19	LED [5]	3.3V
LEDR[6]	PIN_Y19	LED [6]	3.3V
LEDR[7]	PIN_W20	LED [7]	3.3V
LEDR[8]	PIN_W21	LED [8]	3.3V
LEDR[9]	PIN_Y21	LED [9]	3.3V

Signal Name	FPGA Pin No.	Description	I/O Standard
GPIO_0[0]	PIN_AC18	GPIO Connection 0[0]	3.3V
GPIO_0 [1]	PIN_Y17	GPIO Connection 0[1]	3.3V
GPIO_0 [2]	PIN_AD17	GPIO Connection 0[2]	3.3V
GPIO_0 [3]	PIN_Y18	GPIO Connection 0[3]	3.3V
GPIO_0 [4]	PIN_AK16	GPIO Connection 0[4]	3.3V
GPIO_0 [5]	PIN_AK18	GPIO Connection 0[5]	3.3V
GPIO_0 [6]	PIN_AK19	GPIO Connection 0[6]	3.3V
GPIO_0 [7]	PIN_AJ19	GPIO Connection 0[7]	3.3V
GPIO_0 [8]	PIN_AJ17	GPIO Connection 0[8]	3.3V
GPIO_0 [9]	PIN_AJ16	GPIO Connection 0[9]	3.3V
GPIO_0 [10]	PIN_AH18	GPIO Connection 0[10]	3.3V
GPIO_0 [11]	PIN_AH17	GPIO Connection 0[11]	3.3V
GPIO_0 [12]	PIN_AG16	GPIO Connection 0[12]	3.3V
GPIO_0 [13]	PIN_AE16	GPIO Connection 0[13]	3.3V
GPIO_0 [14]	PIN_AF16	GPIO Connection 0[14]	3.3V
GPIO_0 [15]	PIN_AG17	GPIO Connection 0[15]	3.3V
GPIO_0 [16]	PIN_AA18	GPIO Connection 0[16]	3.3V
GPIO_0 [17]	PIN_AA19	GPIO Connection 0[17]	3.3V
GPIO_0 [18]	PIN_AE17	GPIO Connection 0[18]	3.3V
GPIO_0 [19]	PIN_AC20	GPIO Connection 0[19]	3.3V
GPIO_0 [20]	PIN_AH19	GPIO Connection 0[20]	3.3V
GPIO_0 [21]	PIN_AJ20	GPIO Connection 0[21]	3.3V
GPIO_0 [22]	PIN_AH20	GPIO Connection 0[22]	3.3V
GPIO_0 [23]	PIN_AK21	GPIO Connection 0[23]	3.3V
GPIO_0 [24]	PIN_AD19	GPIO Connection 0[24]	3.3V
GPIO_0 [25]	PIN_AD20	GPIO Connection 0[25]	3.3V
GPIO_0 [26]	PIN_AE18	GPIO Connection 0[26]	3.3V
GPIO_0 [27]	PIN_AE19	GPIO Connection 0[27]	3.3V
GPIO_0 [28]	PIN_AF20	GPIO Connection 0[28]	3.3V
GPIO_0 [29]	PIN_AF21	GPIO Connection 0[29]	3.3V
GPIO_0 [30]	PIN_AF19	GPIO Connection 0[30]	3.3V
GPIO_0 [31]	PIN_AG21	GPIO Connection 0[31]	3.3V
GPIO_0 [32]	PIN_AF18	GPIO Connection 0[32]	3.3V
GPIO_0 [33]	PIN_AG20	GPIO Connection 0[33]	3.3V
GPIO_0 [34]	PIN_AG18	GPIO Connection 0[34]	3.3V
GPIO_0 [35]	PIN_AJ21	GPIO Connection 0[35]	3.3V