

# 1 TKgate tutorial

TKgate is a program for simulation electrical and logical circuits, and its manual can be found in the site <http://www.tkgate.org/>.

TKgate has two modes, building and simulations. During the build you can add gates , IO interfaces and ect, and wire them together. During the simulation, the program will “run” your circuit and give you the results.

## 1.1 Beginning

Run tkgate by running konsole and then type tkgate. Choose a new circuit from the file menu.

# 2 Working with a single bit

Read section 2.1 and 2.2 in the manual.

## 2.1 IO

Read sections 2.4 to 2.7 in the tkgate manual.

A single variable input is given by a switch in the make/io menu. (constants are given by the Ground (0) and Vdd (1) in the make/io menu).

Add a switch and a led to you circuit (right click, and select it in the IO submenu). Connect the two together and run the circuit. Next, trim the connection you made (using the cut wire tool).

## 2.2 Gates

Add inverter to your circuit (right click, gates, inverter) and connect it to your switch and to the led. run the circuit.

## 2.3 Debugging and Delay

Read section 2.8.4

Build the circuit :  $X1 \cdot X2' + X2 \cdot X3$ . Also add a led in the end of the circuit.

Double click on the inverter and press on delay. change to custom delay and set the delay value of all the components in the circuit to 10.

Set the names of all the outputs (by double clicking on them) to reasonable names. now run the simulations. Double click on the output of the and, not, and or gates. now set all the values to 1 and press the play. now set the value of x2 to 0. do you see the hazard? Now fix the circuit by adding the term  $X1 \cdot X3$ . That is, build the term  $X1 \cdot X2' + X2 \cdot X3 + X1 \cdot X3$ . In order to implement an or gate with 3 entrances, double click on the gate, goto the port tab and click on the add button. Rerun the circuit. do you see the hazard now?

Note that in combinatorial circuits (like the ones we have) the circuit simulates until no more changes are predicted without changing the inputs again.

## 3 Working with modules

A module in TKgate is equivalent to a sub procedure (or in the course terminology to several Boolean functions). A module consists of inputs, outputs and the implementation.

To build a new module, select the module menu and click on new and give it a name. Remember, the modules names are case sensitive.

### 3.1 Defining module Input and Output

Now we will define the inputs/outputs of the module. Select the edit interfaces option in the module menu.

To add input/output right click on the module side. Define 3 entrances (call them X1,X2,X3) to your module and one exit (call it O1). Now, choose the open option from the module menu.

## 3.2 Connect the ports to the circuit in the module

To connect to the module input/output (called a net) right click and from the module menu choose input/output. Put in the name the name you have chosen for your input/output. (that is X1,X2 or X3 for the input and O1 for the output). Build the circuit from section 2.3 with the inputs of the module instead of the switches and the output of the module instead of the led.

## 3.3 Connecting into your circuit

Double click on the module main at the right hand side. Now left click on the mouse and press B (capital B). Insert the name of the model you chose.

Define 3 switches and a led, and connect the inputs/output. Run the circuit.

# 4 Working with several bits

In TKgate we can define each variable to be a string of bits (instead of 1).

## 4.1 IO

Start a new circuit.

The input for string of bits is called DIP switch. A display for multiple bits is called a LED bar or a 7-seg LED (Dec) and 7-seg LED (Hex). Add a DIP switch, a led bar and two 7-seg LEDS to your circuit and connect them together. Run the circuit. Click on the dip switch while running and change the value of the DIP switch. The value are in hexadecimal code (for bunches of 4 bits). that is aa = 10101010 , ff= 11111111 and 20 = 00100000. stop the simulation. You can change the width of the DIP switch or the LED bar by double clicking on them and setting, in the port section, the number of bits.

## 4.2 bitwise Gates

The regular gates are operating bit-wise. That is, on two strings of bits, A=abcd and B=wxyz the result of AND(A, B) is the bit string (a\*w)(b\*x)(c\*y)(d\*z). Add another

DIP switch to the circuit. Add an AND Gate to the system. Change the input and output width of the and gate to 8 bits. connect the two DIP switches to the and gate and connect it to a LED bar. Run the simulation.

### 4.3 Reduction gates

Reduction gates of the bits string operate on a single bit string and perform the operation on the bits themselves. for example, Reduction AND of  $A = abcd$  is  $a*b*c*d$ . Note that the result is a single bit. Add a reduction gate to the system and build a circuit with Reduction XOR and a led.

### 4.4 Reducing multiple bits to 1 and expanding 1 to multiple bits

It is sometimes useful to check a single bit, or to connect several single bits into a string.

The most useful way to convert n-bit string to k substrings (not necessarily the same width) is with the wire merge tool (in the IO menu).

Start a new circuit. Add a DIP switch with size 3. Add a wire merge tool and connect the right exit to the DIP Switch. double click on the left green output (numbered 1). Change the bit width to 2. add a LED bar and connect it to the line you just changed. Run the circuit. As you can see, the wire merge gives you direct access to the first two bits of the bit string. This is bi-directional. To see it, add two switches (of 1 bit) to the circuit, a wire merge and a LED bar. now connect the switches to the green lines and the led bar to the red bar. Run the circuit.

To inspect a single bit, connect a led (of single bit) to the output of a dip switch. The number on the connection denote the bit location that the led is tapped into. (that is 3 is the the third bit or when the value is 8). Change the value by double clicking on the connection.

## 5 Some built in circuits

Some operations are encountered frequently and therefore the circuit that performs the calculations is already available as a chip you can buy. In tkgate, some circuits are already implemented in the software. The important ones are adder, multiplexer and decoder. Read on the adder, multiplexer and decoder in section 2.16.

### 5.1 Adder

build a circuit that adds two 3 bit numbers. To see the result define a 7-seg led that shows decimal numbers.

### 5.2 Multiplexer

Build a small circuit with multiplexer to make sure you understand what are its functions.

### 5.3 Decoder

Build a small circuit with multiplexer to make sure you understand what are its functions.

## 6 Building subtractor using 2 complement

Build a circuit that subtracts two numbers with 3 bits using the 2-complements. To do so, use two adders, and inverter gate, three DIP switches and a single LED.

## 7 What you should know

You should know how to

- work with single and multiple bits.
- know all the options from the menu make/IO (except for clock and tty), make/gate make/reduction what a decoder from the make/MSI what is adder from the make/ALU and what the different options in the

- Make sure to read the rest of the manual. Some things are important for debugging (like finding a gate).