BASIC FEATURES

Standard Components

There is a pallet of components on the left side of each circuit window; these can be used in the simulation by dragging and dropping with the mouse in the usual way.

he ground and Vcc supplies at the top of the pallet are constant logic(0) and logic(1) sources respectively. When the simulation is running, a single mouse click on a switch inverts its output from logic(0) to logic(1) and the next click switches it back to logic(0). The button output normally stays at logic(0) but will change to logic(1) as long as the mouse is held down over it. There is a choice of two output devices: the single LED which glows red when its input is at logic(1), and the Display4 which decodes the levels on its inputs and displays the result as a hexadecimal (0-9,A-E) digit on a seven-segment display. Input4 is a convenient way of generating hexadecimal input. Clocks generate square waves, alternating between logic(0) and logic(1), a clock's period, in simulation time units, can be adjusted by selecting it and typing âŒ˜-T which invokes the component options dialogue. Clicking a clock with the mouse will start, or stop, the clock running. The Beeper plays a sound that can be changed with the component options dialogue where there is also an option to beep only in response to rising edges.

he table lists the two-input gates and their state tables, X represents an indeterminate state (see next section). All the gates listed in this table (and some of the buffers) can be rotated to point in directions other than the usual West-East by holding âŒ˜-shift and clicking them one or more times.

Logic States and Buffers

Each pin in the simulated circuit is assigned one of four states. Two of these, logic(1) and logic(0), are just the usual binary values; an output at either of these levels determines level of all the connected inputs. A single input must not therefore be connected to two outputs, they might have different values and the outcome would be indeterminate. LogicSim does not allow such connections to be made accidentally but the indeterminate, or "Don't Know", state can arise in other ways and is given the value logic(X).

There are circumstances, for example when different parts of a system communicate via a bus, when one really does want to have several physically separated outputs driving the same line. This can be accomplished by using the Transistor which is one of the buffers.

ogicSim has three kinds of buffer. The plain Buffer simply sets its output equal to its input after the appropriate delay and the output of the NOT buffer is similarly the negated value of its input. The Transistor buffer, also known as a 3-state buffer, connects its input direct to its output if the base connection is at logic(1) or, when the base is at logic(0), disconnects

the output completely and leaves it in the high-impedance state logic(Z). Logic(Z) is regarded as an indeterminate input by all devices.

Two or more outputs can be connected together, but only at a node created on a connecting wire with by âŒ˜-option-Click. Such cicuits must be designed to avoid outputs at different, or indeterminate levels, being connected together unless one of them is in the logic(Z) state. If a cicuit breaks this rule the affected gates will catch fire. To see which gates connected to a bus have logic(Z) outputs use the Show internal values option from the Simulation menu.

When LogicSim 2.7 starts a simulation it normally assumes that all states are initially indeterminate and sets them to logic(X). This performs the useful check that during operation the state of the circuit is a function solely of its inputs and time, not of the initial state. The default initial state can be set to other values using the Options... dialogue from the Simulate menu, but you probably don't want to do this unless you have a very good reason. One good reason is that circuits involving outputs connected together will always catch fire with logic(X) as the initial state so logic(Z) should be used instead in these cases.

Flip-Flops

LogicSim 2.7 provides several types of flip-flop:

SH is a simple set-reset flip-flop and is clocked by a positive-going edge at the H input. The most flexible of the supplied flip-flops is JK2 a positive-edge triggered JK-type with asynchronous preset and clear inputs. JK is a simpler version, equivalent to a JK2 with its preset and clear terminals tied to Vcc internally. The D-type, or "data", flip-flop DFF has only one input D, which is transferred to the output on receipt of a positive-going edge at the clock input. DLATCH is very similar device that, instead of being edge triggered, transfers its D input to the output while the clock input is at logic(1). The T-type or "toggle" flip-flop TFF also only has one input; when T=logic(1) the next positive-going edge at the clock toggles the output. Before using these components do check that they follow the conventions you expect. For example, DFF is a positive-edge triggered D-type flip-flop and the example file "Faulty Pulse Synchroniser" illustrates its use in a circuit that requires a negative-edge triggering.