Synopsis - Week 5 Integrated Project: Universal NAND gates, FSM counter, and Boolean Logic

Prof. Jordan C. Hanson

September 30, 2021

1 Universal NAND Gates

1. Derive a two-input AND gate from NAND gates. (a) Write this algebraically. (b) Draw the gates below.

2. Derive a three-input AND gate from NAND gates. (a) Write this algebraically. (b) Draw the gates below.

3. Create a blank Jupyter notebook. Use the `boolean_generator` code to create a three input AND gate from NAND gates. Set the inputs to the push buttons and the output to an LED (as usual).

2 FSM Counter

1. Recall the FSM counter in a prior laboaratory activity. Create an FSM that acts as a 3-bit binary counter. Assume the direction is always "forward," so there is no need for a direction input. Add this code to your Jupyter notebook. Make sure to reset the kernel when new code is added, to ensure overlays are downloaded to the PL correctly.

2. Verify that the FSM is counting correctly by using the `trace` and `show_waveform` methods of the `fsm_generator`. The trace method has to be run first.

3 Stepping from FSM to Boolean

1. Recall from last time that the stepping controller can take the PL from one set of digital functions to the next. See the `single_stepping_generator` example for further details.

2. Adapt the `single_stepping_generator` code into your Jupyter notebook so that the three-input AND gate, constructed from NAND gates, receives the output of the FSM counter.

3. Does the LED go HIGH when it should? Verify with the trace and timing diagram functionality that the behavior is correct.