Lab 1 COMPENG 3DQ5

Joshua Umansky, George Wahba

400234265, 400371904

McMaster University

COMPENG 3DQ5

Lab 1 COMPENG 3DQ5

To implement the logic behaviour described in the lab report for the green LEDs, we used K-maps and determined the logic elements that could be used to achieve different outputs. For LED 8, since it only required **one** switch between 10 to 16 to be high on the board, this implies that we could use a *OR* logic element to represent that behavior. LED 7 requires that **all** switches between 10 to 16 be high on the board, therefore it is clear that we use an *AND* operator to implement the desired logic. LED 6 asks that an **even** number of switches between 16 and 10 are high to be lightened, we used a *NXOR* gate to implement it, because an XOR gate is high output when there is an odd number of inputs high, therefore "not-ing" it would give us the desired result. A *NAND* gate was used for LED 5 because it asks that at **least one** of the switches between 9 to 3 is low, which means that it only won't turn on when all switches are high, meaning that reversing an *AND* gate logic will result in the desired output. An *AND* was used for LED 4 because it is lightened when only **none** of the switches between 9 and 3 are low, or to reverse the statement, when **all** the switches between 9 and 3 are high. LED 3 is lightened when the number of switches that are low between 9 and 3 are **even** or, when the number of switches between 9 and 3 that are high are **odd**, meaning that we can use an *XOR* gate to implement the logic. LED 2, LED 1, and LED 0 were implemented via a more "brute force" method where we just added each Switch 0, Switch 1, and Switch 2 together and used different comparators to achieve our desired output. When designing the hex-7 seg. display converter, we first set up two separate methods for converting to hex and octal, based on how many bits are required for each number system, afterwards, a Boolean value would be assigned to represent either hex or octal being shown on the screen, and if hex was chosen, the last segment of the displays would be blank since hex is only requiring 2 "digits" to represent 8, and if octal was chosen, since the total digits is 8, meaning that there would be 2 "hanging" digits, a separate conversion system was created to represent the 4 possible combinations of 0 to 3 that the last 2 bits of the display could take.