Blackstone Valley Regional Vocational Technical High School
Student Portfolio- Project Reflection

Date	**10-19-23**
Student Name	**Om Patel**
Subject	**Date of Birth - DE (11th)**
Instructor's Name	**Mrs. dos Santos**
Instructor's Signature	**Mrs. dos Santos**

In this project I was challenged to build a display that showcases a person's birthday so people can compare them, testing the "Birthday Paradox." In this project, we are only allowed to use 2 input gates creating circuits using, NAND, NOR, and AOI logic. We must use at least one circuit that is completely made of NAND logic and one circuit that is made up of NOR logic. We are also informed to use a technique called Karnaugh mapping to derive simplified expressions.

In the first part of this project, I created a truth table to identify to logic circuits that I must create for the project. To create this truth table, I used binary number counting from zero to seven to fill out all the possible input combinations. I also used the knowledge of how a seven-segment display works to find the eight possible outputs. A seven-segment display uses one of seven inputs to light a singular segment. As an example to make a 2, I would not light segment F and segment C, which I would do by not powering port F or C. Once I fill out my truth table I can then create a Karnaugh map or K-Map for all seven segments to get a simplified expression. To use K Mapping to simplify you must first begin by labeling and constructing a Karnaugh Map. Next, fill in the cells that correspond to the 1s in the truth table with 1s, and place 0s in all the other cells. Afterward, identify and group isolated 1s, which cannot form larger groups. Then, proceed to group any hex, octet, quad, and pair configurations, even if they contain some 1s already grouped but not enclosed in a larger group. Finally, add all the grouped terms to generate the final equation. The reason why I needed a simplified expression as it allows me to eliminate any repetition in my circuit, making it more efficient and more cost-effective. After getting my logic expressions for my 8 outputs I then identified the two circuits I wanted to make either NAND or NOR-only circuits. With my logic expressions made, I could then start my MultiSIM simulation and sketches. I first created the gates that were repeated making the expressions their own subcircuits that I will use for my other AOI circuits. The reason why I did this is because it allows me to use fewer gates and make my circuit more efficient. With my MultiSIM schematic made for my AOI logic, I then started to sketch out my NAND and NOR circuits. To create a circuit using either NAND or NOR gates one must first create a circuit using AOI logic and then identify each AND, OR, and Inverter. Once identified you must replace the gate with its NAND or NOR counterparts and then redraw your circuit. Then you have to identify any double inversions and take them out. With them out, you can redraw your final circuit. The reason why I created these specific circuits as NAND and NOR circuits is that the creation of these circuits in AOI logic is costly and slower as it would add just an excess of ICs. Whereas the use of one IC type is cheaper and faster. The reason why I used 5 AOI circuits is because of the repeated use of my 4 AND gates and inverters. This allows me to cut down on the number of gates I use, whereas on a NAND or NOR I would have to use 3 different gates to invert each signal. With all my subcircuits created, I could then attach each to its corresponding port on the seven-segment display and complete my simulation.

With my MultiSIM complete I could then use it as a schematic to create my TinkerCAD. TinkerCAD is a digital breadboard software used to visualize what a physical breadboard of a circuit would look like. This is used as it is like breadboarding without the repercussion of physically making it, as I could have infinite possibilities to create breadboard prototypes. It also saves on material, as if I make a mistake in a digital version, I can easily just delete a wire. In my TinkerCAD, I started by first identifying the amount of breadboards I would need, then connecting the power and ground power for each board. I would then place the 3 switches I would need for inputs X, Y, and Z. After I placed down my ICs on the breadboard, identifying each breadboard to a type of gate, I then placed the ground and power for my switches and chips. I then placed my seven-segment display and started placing all my vertical wires. The vertical wires are long strands of wire that connect a signal in a column, in multiple breadboards. This idea came from the bus tool in MultiSIM which is a node where different multiple power lines are connected, however instead of having multiple power lines each bus, or vertical wire line would hold one signal. While making my circuit I also created a color coding key that identified what each wire color signal was. Once I finished placing my busses I then created inverted signals for all three of my inputs, in which I created more vertical wires. This then allowed me to create the 4 AND gates that are used repeatedly throughout my AOI logic circuits, and then create a bus for each. After this I could start the rest of my circuit, connecting my inputs, chips, and outputs into one another, to then create one output wire that I would place into my seven-segment display. I did this same process for my NAND and NOR logic gates.

With my TinkerCAD complete I then started to create my physical prototype on a breadboard. Using the TinkerCAD as a schematic I gathered my components. While breadboard I continued to use a color-coordinated system to wire my circuit and followed the same steps I used to create the schematic itself. After placing down my components, I then placed any ground and power cables and created my inverted signals from my inverter IC. After, I created the gates that are used repeatedly throughout my AOI logic circuits and then continued to wire the rest of my components alphabetically based on each individual sub-circuit avoiding the NAND and NOR circuits. With my AOI circuits complete I decided to test what I had placed down so far. In doing so I counter one small issue which was that I placed my USB which powered my board into a faulty USB port. What was confusing though was that it still indicated that the board was on as the myDAQ's light was on. To identify the board wasn't getting power I used a multimeter and noticed that nothing was getting power. After changing the USB port into a different socket I tested my circuit and found that it works. This means I could then start my next circuit in which I built my NAND circuit and sequentially, my NOR. With all my subcircuits built I connected it to power and checked if it worked using the truth table I made. With my circuit working and displaying the date I wanted I had completed the project.

Ultimately, this project taught me the importance of planning when it comes to designing complex circuits. It also taught me how to create efficient and cost-effective circuits and allowed me to practice my skills in deriving an equation using K-Mapping. This project helped me enhance my technical knowledge and value my abilities in precision and problem solving.

Blackstone Valley Regional Vocational Technical High School
Student Portfolio- Project Reflection

Technical competencies and academic skills demonstrated by completing this assignment.

Framework Standard	Description
2.C.03.02	Construct a circuit.
2.C.02.23	Use schematics and symbolic algebra to represent digital gates as part of a solution to a design problem (logic symbols: AND, OR, NOT, NAND, NOR, XOR and X-NOR gates).
2.C.02.24	Create Boolean expressions and truth tables.
2.C.02.27	Formulate and use a Karnaugh Map and/or Boolean algebra to reduce logic equation.
2.C.03.01	Simulate a circuit.
Embedded Academics	**Description**
2.C.02.23	Determine the meaning of symbols, key terms, and other domain specific words and phrases as they are used in a specific scientific or technical context relevant to grades 6–12 texts and topics.
2.C.06.1.5	Interpret plans, diagrams, and working drawings in the construction of prototypes or models
2.C.02.27.2	Create equations in two or more variables to represent relationships between quantities.
2.C.20.9-10	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text