|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Digital Electronics** | **Semester:** | **III** |
| **Date of Performance:** |  | **Batch:** | **A - 3** |
| **Faculty Name:** |  | **Roll no.:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_ / 25** |

**Experiment no.: 5**

**Title: Multiplexer**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| * Implementation of MUX using MSI IC * Implementation of function using decoder IC |

|  |
| --- |
| **COs to be Achieved:** |
| **CO2:** Design combinational logic circuits using MSI devices. |

|  |
| --- |
| **Theory:** |
| **Multiplexers:**  A multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2n inputs has n select lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a data selector.  **Decoder:**  A decoder is a combinational circuit with n inputs and at most 2n outputs. They may be used to route input data to a specified output line, as, for example, is done in memory addressing, where input data are to be stored in (or read from) a specified memory location. They can be used for some code conversions. Or they may be used for data distribution i.e., demultiplexing. |

|  |  |
| --- | --- |
| **Circuit Diagram/Block Diagram:** | |
| http://www.toves.org/books/comps/mux-circ.png  **4:1 Multiplexer** | |  |  |  | | --- | --- | --- | | **S1** | **S0** | **Q** | | 0 | 0 | d00 | | 0 | 1 | d01 | | 1 | 0 | d10 | | 1 | 1 | d11 |   **Q = S1’S0’d00 + S1’S0d01 + S1S0’d10 + S1S0d11** |
|  |  |

|  |
| --- |
| **Stepwise – Procedure:** |
| 1. **Implement a function F (A, B, C) = ∑m (0, 1, 3, 7) using single 8:1 mux.**    1. Case 1: setup board (board off)    2. Case 2: when ∑m (0)    3. Case 3: when ∑m (1)    4. Case 4: when ∑m (3)    5. Case 5: when ∑π (2)              1. **Implement a function F (A, B, C) = ∑m (0, 3, 4, 7) using single 4:1 mux and additional gates if required.**    1. Case 1: setup board (board off)    2. Case 2: when ∑m (0)    3. Case 3: when ∑π (2)    4. Case 4: when ∑m (7)    5. Case 5: when ∑π (1)              1. **Implement 3 to 8 decoder for the given function (will be given by the faculty).** |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. **Implement full adder Sum(A, B, Cin) and Cout(A, B, Cin) with**    1. **Two 8:1 multiplexers**    2. **Two 4:1 multiplexers** 2. **There are four adjacent parking slots in a company. Each slot is equipped with a sensor whose output is asserted low when a car is occupying a slot, otherwise sensor output is high. Design and draw a schematic for a system, which will generate a low output if and only if there are two or more than two adjacent slots vacant.**   **Implement it using 8: 1 mux.**   1. **Implement F(A, B, C) = ∑m (0, 1, 3, 7) using 74138 IC and two input NAND gates.** |

|  |
| --- |
| **Conclusion:** |
| We have effectively employed the 8x1 multiplexer IC-74141 and the 4x1 multiplexer IC-14153. Furthermore, we have learned how to utilize both the 8x1 and 4x1 multiplexers to create a full adder. The decoding process using IC-74138 has been successfully implemented, yielding the desired outcomes observed in the above cases. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |