Question 2

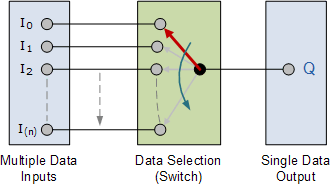
1. The multiplexer is a combinational logic circuit designed to switch one of several input lines to a single common output line

Multiplexing is the generic term used to describe the operation of sending one or more analogue or digital signals over a common transmission line at different times or speeds and as such, the device we use to do just that is called a **Multiplexer**.

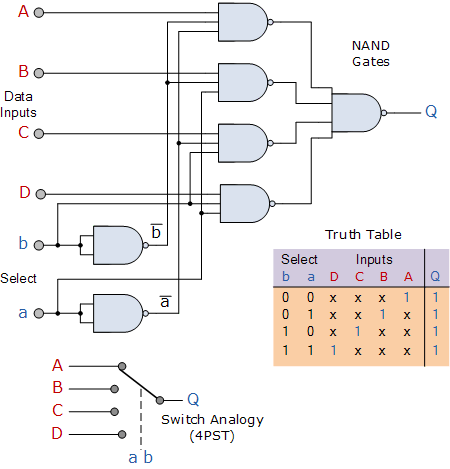
The multiplexer, shortened to “MUX” or “MPX”, is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal. Multiplexers operate like very fast acting multiple position rotary switches connecting or controlling multiple input lines called “channels” one at a time to the output.

Multiplexers, or MUX’s, can be either digital circuits made from high speed logic gates used to switch digital or binary data or they can be analogue types using transistors, MOSFET’s or relays to switch one of the voltage or current inputs through to a single output.

**Basic Multiplexing Switch**



### 4:1 Multiplexer



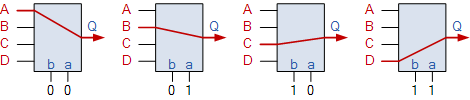
The Boolean expression for this 4-to-1 **Multiplexer** above with inputs A to D and data select lines a, b is given as:

Q = abA + abB + abC + abD

In this example at any one instant in time only ONE of the four analogue switches is closed, connecting only one of the input lines A to D to the single output at Q. As to which switch is closed depends upon the addressing input code on lines “a” and “b“.

So for this example to select input B to the output at Q, the binary input address would need to be “a” = logic “1” and “b” = logic “0”. Thus we can show the selection of the data through the multiplexer as a function of the data select bits as shown.

### Multiplexer Input Line Selection



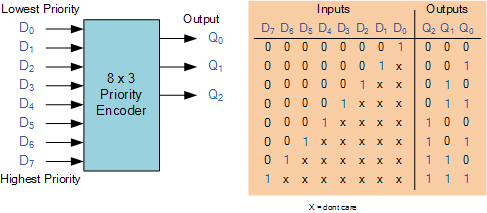
Adding more control address lines, (n) will allow the multiplexer to control more inputs as it can switch 2n inputs but each control line configuration will connect only ONE input to the output.

Then the implementation of the Boolean expression above using individual logic gates would require the use of seven individual gates consisting of AND, OR and NOT gates as shown.

1. The **Priority Encoder** solves the problems mentioned above by allocating a priority level to each input. The priority encoders output corresponds to the currently active input which has the highest priority. So when an input with a higher priority is present, all other inputs with a lower priority will be ignored.

The priority encoder comes in many different forms with an example of an 8-input priority encoder along with its truth table shown below.

**8-to-3 Bit Priority Encoder**



Priority encoders are available in standard IC form and the TTL 74LS148 is an 8-to-3 bit priority encoder which has eight active LOW (logic “0”) inputs and provides a 3-bit code of the highest ranked input at its output.

Priority encoders output the highest order input first for example, if input lines “D2“, “D3” and “D5” are applied simultaneously the output code would be for input “D5” (“101”) as this has the highest order out of the 3 inputs. Once input “D5” had been removed the next highest output code would be for input “D3” (“011”), and so on.

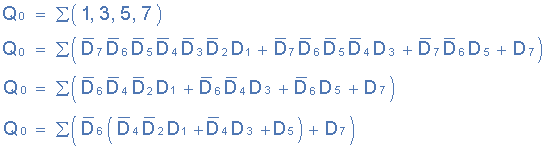
The truth table for a 8-to-3 bit priority encoder is given as:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Digital Inputs | | | | | | | | Binary Output | | |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Q2 | Q1 | Q0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | **1** | X | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | **1** | X | X | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | **1** | X | X | X | 0 | 1 | 1 |
| 0 | 0 | 0 | **1** | X | X | X | X | 1 | 0 | 0 |
| 0 | 0 | **1** | X | X | X | X | X | 1 | 0 | 1 |
| 0 | **1** | X | X | X | X | X | X | 1 | 1 | 0 |
| **1** | X | X | X | X | X | X | X | 1 | 1 | 1 |

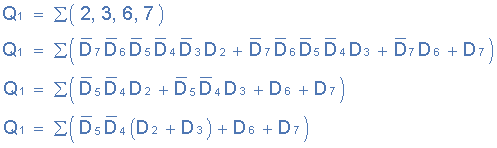
Where X equals “dont care”, that is logic “0” or a logic “1”.

From this truth table, the Boolean expression for the encoder above with data inputs D0 to D7 and outputs Q0, Q1, Q2 is given as:

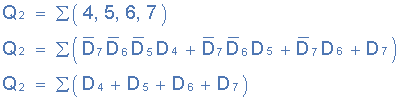
Output Q0



Output Q1

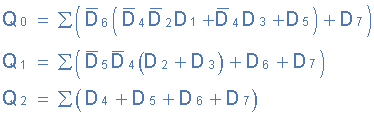


Output Q2



Then the final Boolean expression for the priority encoder including the zero inputs is defined as:

**Priority Encoder Output Expression**



In practice these zero inputs would be ignored allowing the implementation of the final Boolean expression for the outputs of the 8-to-3 **priority encoder**.

Links: <https://www.youtube.com/watch?v=kEj-m3YuGa4>

: <https://www.youtube.com/watch?v=g1Lfz1XgrH8&t=8s>