

INTRODUCTION TO COMPUTER SYSTEMS (IT1020)

Year 1, Semester 1

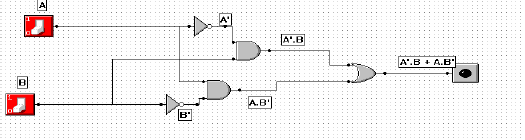
Work Sheet 05

IT22278944, Vikum Chathuranga Basnayaka

Y1.S1.WD.IT.17

11-12-2022





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A’ | B’ | A’. B | A.B’ | F |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |

a.)

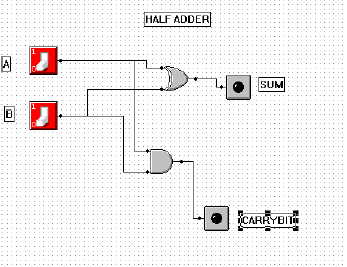


|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1. The resemblance of the XOR circuit to the circuit

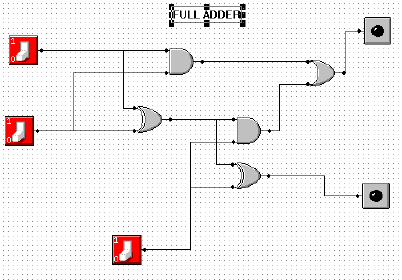
* Light is turned on when one switch is turned on.
* Off when both switches are turned on and off when both switches are turned off.

1. Half adder



|  |  |  |  |
| --- | --- | --- | --- |
| A | B | SUM | CARRYBIT |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

1. Full adder

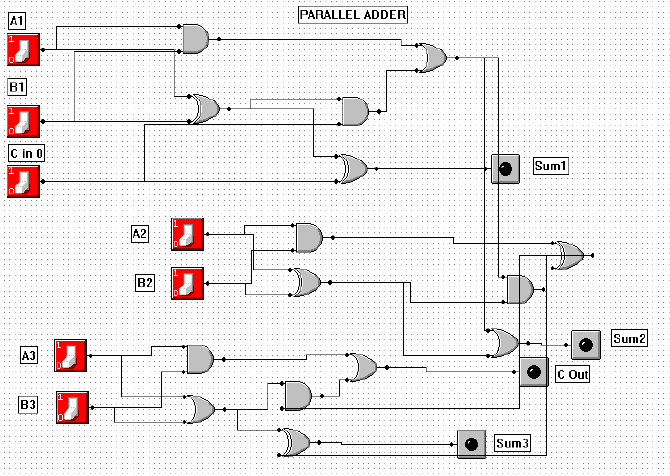


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | SUM | CARRYBIT |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

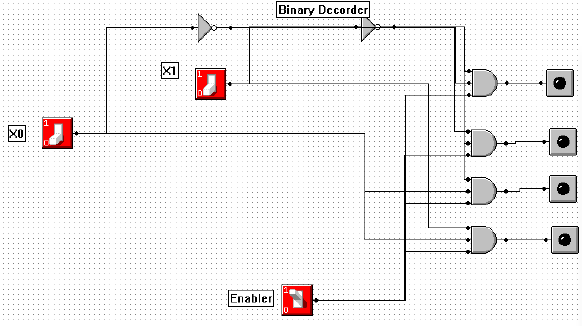
The purpose of these circuits is,

* A category of combinational logic circuits known as the Half Adder adds two 1-bit binary digits. It produces the carry as well as the sum of the two inputs. The Full Adder is another a kind of combinational logic that performs addition operations by adding three of the 1-bit binary digits.

1. Parallel adder

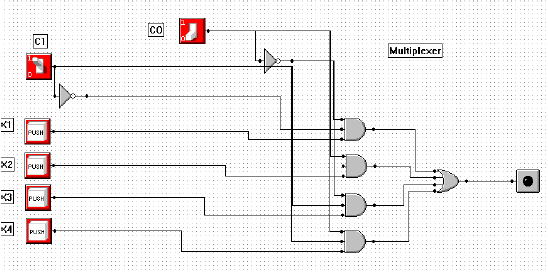


1. Binary decoder



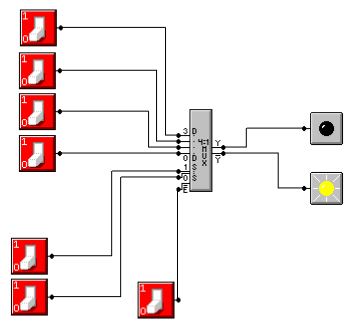
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | ENABLER | Y0 | Y1 | Y2 | Y3 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

1. Multiplexer



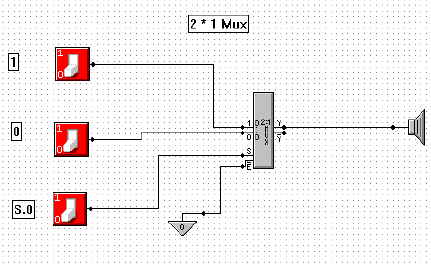
|  |  |  |  |
| --- | --- | --- | --- |
| C1 | C2 | X | M |
| 0 | 1 | 0  X1  1 | 0  1 |
| 0 | 0 | 0  X2  1 | 0  1 |
| 1 | 1 | 0  X3  1 | 0  1 |
| 1 | 0 | 0  X4  1 | 0  1 |

1. 4 to 1 mux



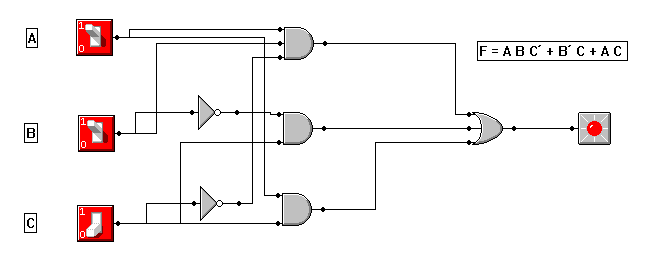
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S0** | **S1** | **D0** | **D1** | **D2** | **D3** | **Y** |
| 0 | 0 | 0 | X | X | X | 0 |
| 0 | 0 | 1 | X | X | X | 1 |
| 0 | 1 | X | 0 | X | X | 0 |
| 0 | 1 | X | 1 | X | X | 1 |
| 1 | 0 | X | X | 0 | X | 0 |
| 1 | 0 | X | X | 1 | X | 1 |
| 1 | 1 | X | X | X | 0 | 0 |
| 1 | 1 | X | X | X | 1 | 1 |

1. 2 to 1 mux



|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | S0 | **Y** |
| 0 | 0 | X | 0 |
| 0 | 1 | X | 1 |
| 1 | X | 0 | 0 |
| 1 | X | 1 | 1 |

1. F = A B C’ + B’ C + A C



|  |  |  |  |
| --- | --- | --- | --- |
| Input | | | Output |
| A | B | C | F = A B C’ + B’ C + A C |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |



