CSE306 (Computer Architecture Sessional)

A Report On

**4-bit ALU Simulation**

Contributor Student ID:

1705068

1705069

1705070

1705071

1705074

Section: B

Subsection: B1

Group NO: **3**

Introduction:

Problem Specification:

We are to design a 4 bit-ALU Circuit as well as show the effects of various operations on flags as per the rules of Assembly Language.

Required Flags:

– Carry (C)

– Sign (S)

– Overflow (V)

– Zero (Z)

Assigned Instructions:

Inputs: A (4-bit)

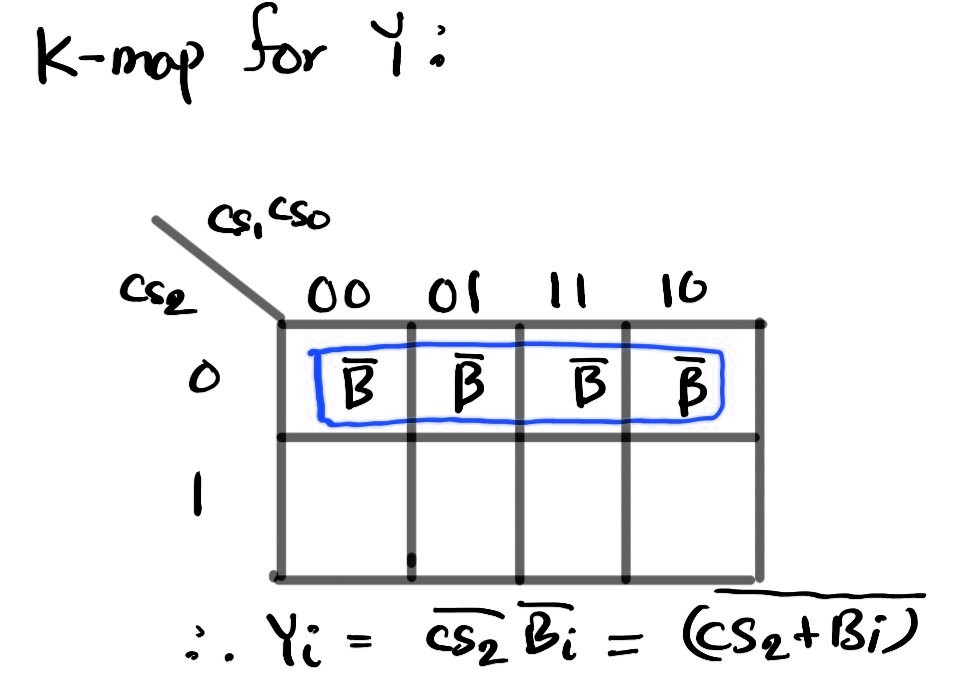
B (4 bit)

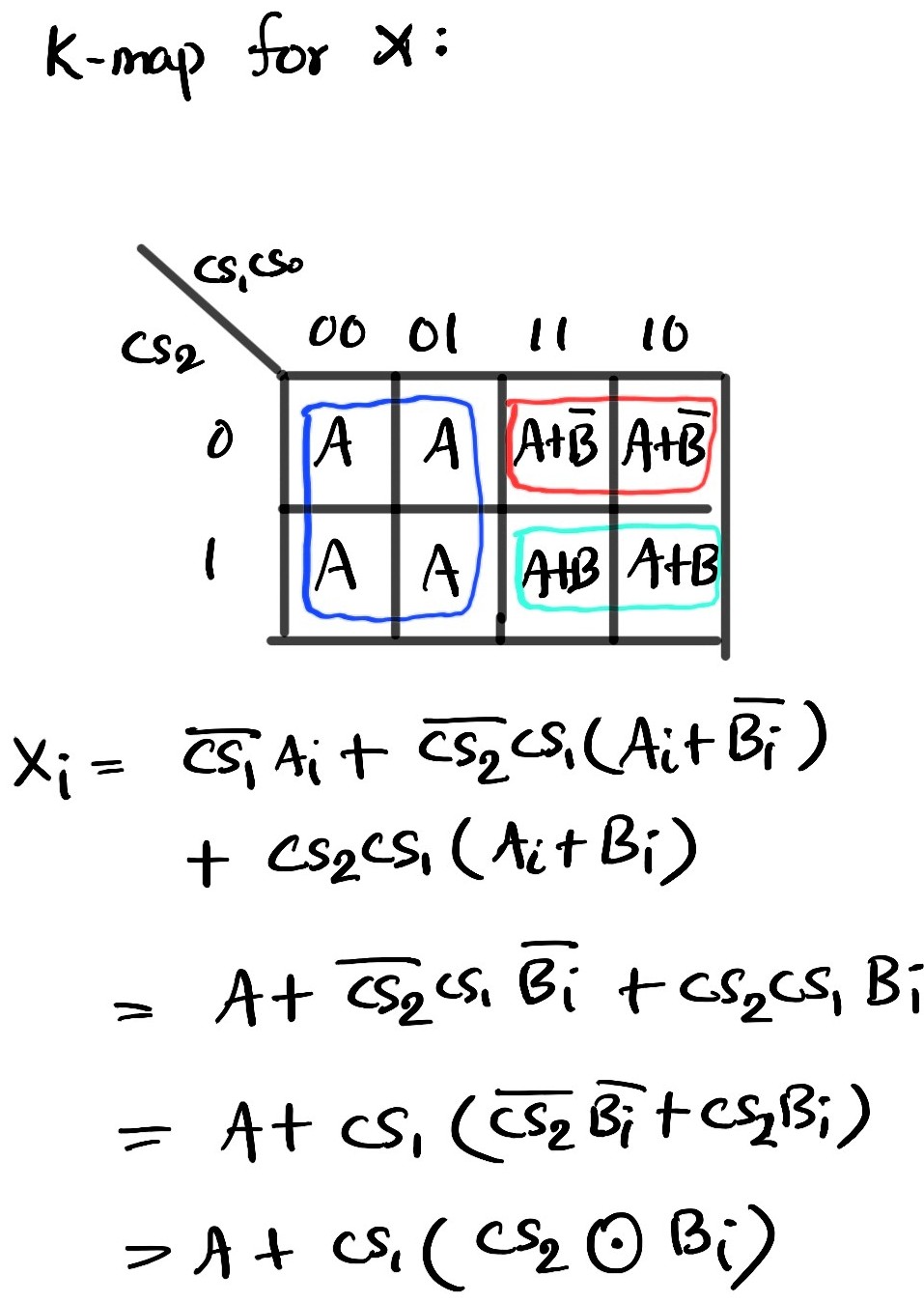
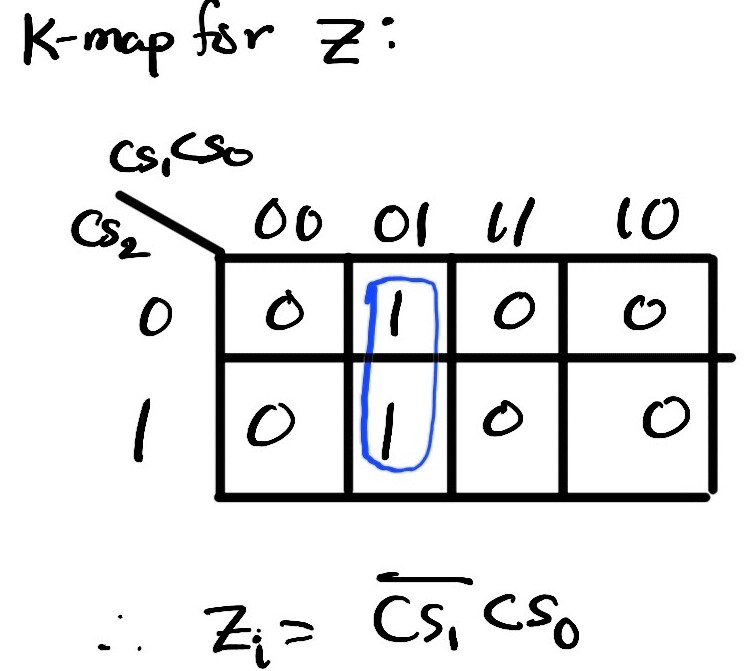
cs1, cs2, cs0 (Selection bits)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cs2** | **cs1** | **cs0(cin)** | **Functions** | **Form** |
| 0 | 0 | 0 | Subtract with borrow | **A +** |
| 0 | 0 | 1 | Subtract | **A + + 1** |
| 0 | 1 | x | AND | **A ∧ B** |
| 1 | 0 | 0 | Transfer A | **A** |
| 1 | 0 | 1 | Increment A | **A + 1** |
| 1 | 1 | x | OR | **A** [**∨**](https://en.wikipedia.org/wiki/Vel_(symbol)) **B** |

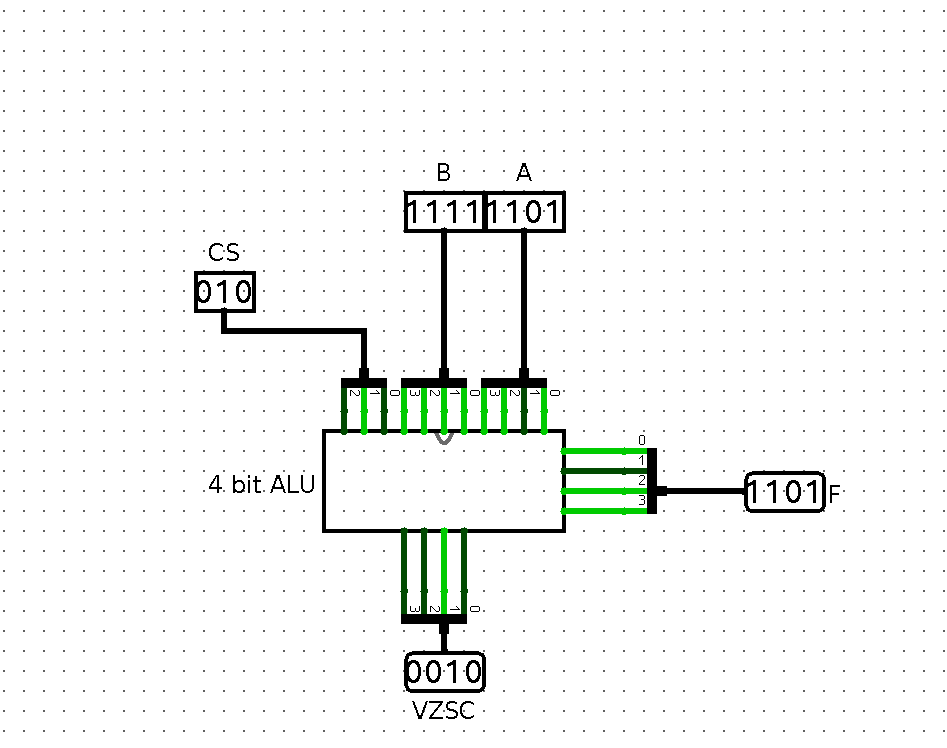
Truth Table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **cs2** | **cs1** | **cs0(cin)** | **Function** | **F** | **X** | **Y** | **Z** |
| 0 | 0 | 0 | Subtract with borrow | **A +** | A |  | 0 |
| 0 | 0 | 1 | Subtract | **A + + 1** | A |  | 1 |
| 0 | 1 | x | AND | **A ∧ B** | A + |  | 0 |
| 1 | 0 | 0 | Transfer A | **A** | A | 0 | 0 |
| 1 | 0 | 1 | Increment A | **A + 1** | A | 0 | 1 |
| 1 | 1 | x | OR | **A** [**∨**](https://en.wikipedia.org/wiki/Vel_(symbol)) **B** | A + B | 0 | 0 |

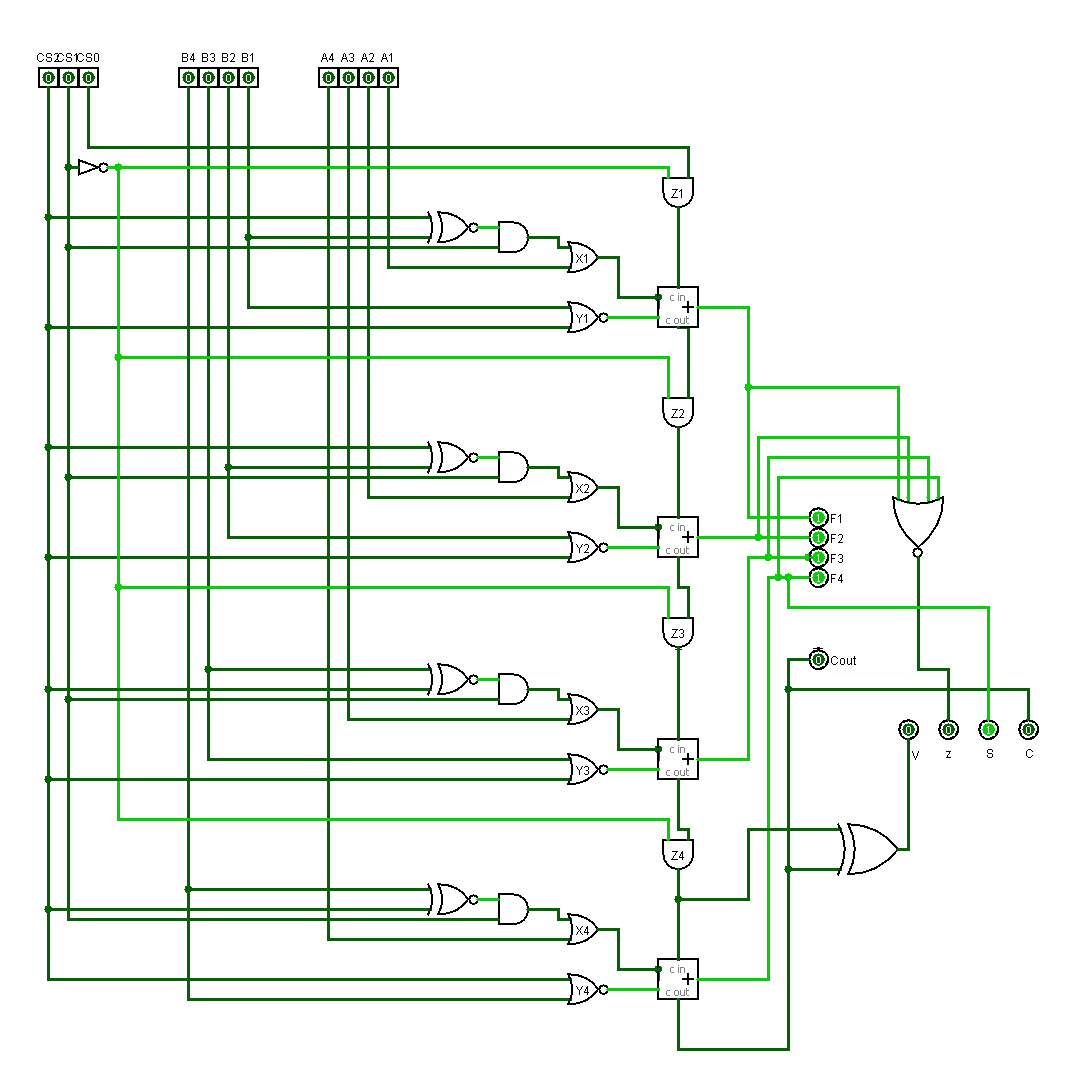




Block Diagram



Circuit Diagram

IC Used with Count:

|  |  |  |
| --- | --- | --- |
| IC | IC Name | Count |
| 7402 | Quad 2-NOR Gate | 1 |
| 7404 | Hex Inverter | 1 |
| 7408 | Quad 2-AND Gate | 2 |
| 7425 | Dual 4-NOR Gate | 1 |
| 7432 | Quad 2-OR Gate | 1 |
| 7480 | 1-Bit Full Adder | 2 |
| 7486 | Quad 2-Exclusive OR Gate | 1 |
| 747266 | Exclusive-NOR Gate | 1 |
|  |  | Total = 10 |

Discussion: While implementing the specified design of the ALU, a minimum number of ICs were used. We used various types of gates such as XOR, NOT, 2 input NOR, 4 input NOR, XNOR, AND, OR and 1 bit Full Adder. In order to check the status bits after the specified operations, a status register was also implemented.