# Homework 1

Zach Neveu

January 15, 2020

1. Problem 1

```
def divide_by_three(integerlist):
    """
    This function outputs a list where, if the integer on
    the input list is divisible by three, the corresponding
    output is 1 else the output is 0
    """
    return [integerlist[i]%3 == 0 for i in range(len(integerlist))]
```

2. Problem 2: full adder

Table 1: Truth table of full adder

| Input 0 | Input 1 | $C_{in}$ | Sum | Carry |
|---|---|---|---|---|
| 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 |

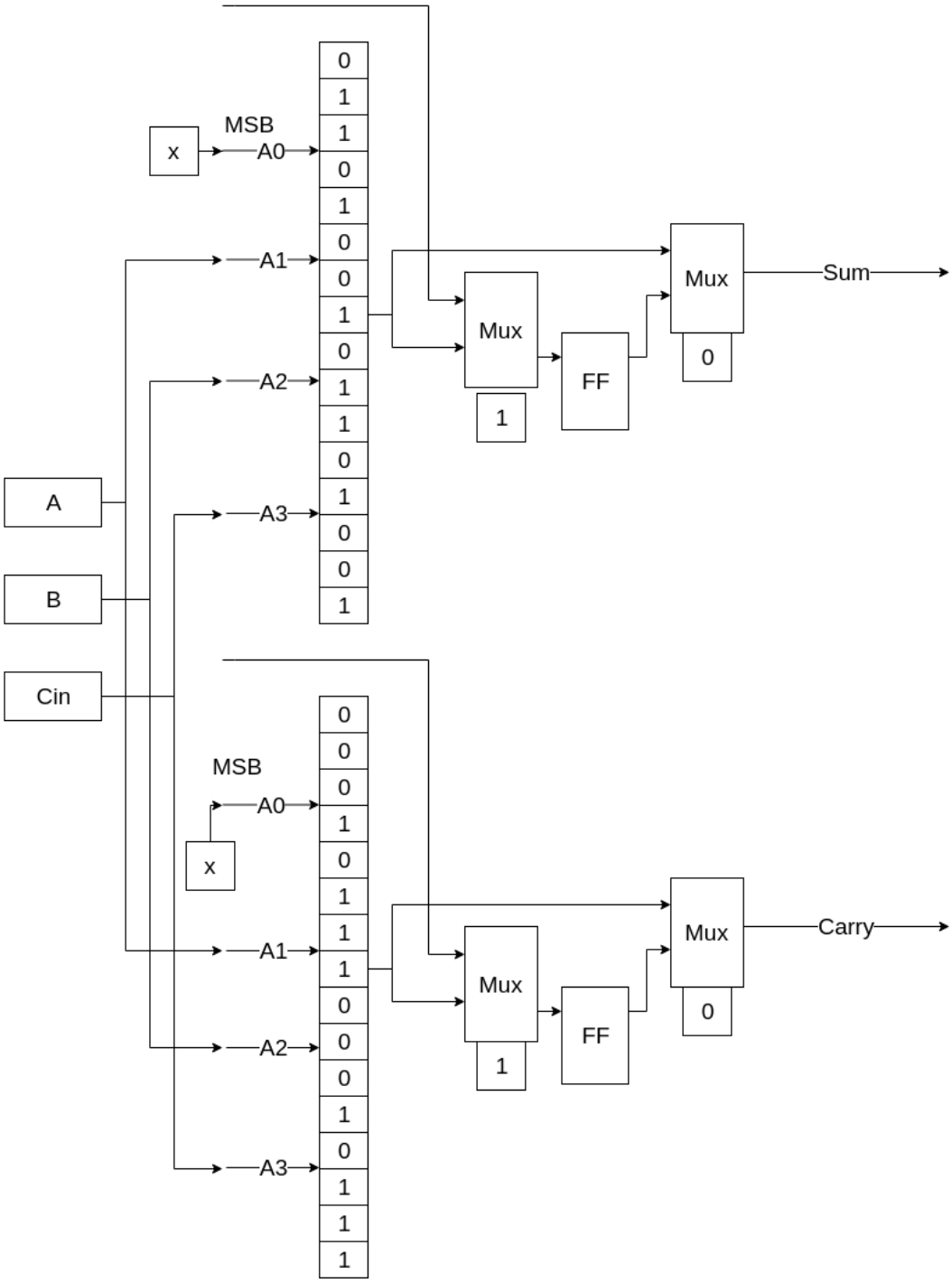


Figure 1: Full Adder on Simplified Logic Slices

3. Problem 3: "101" Detector

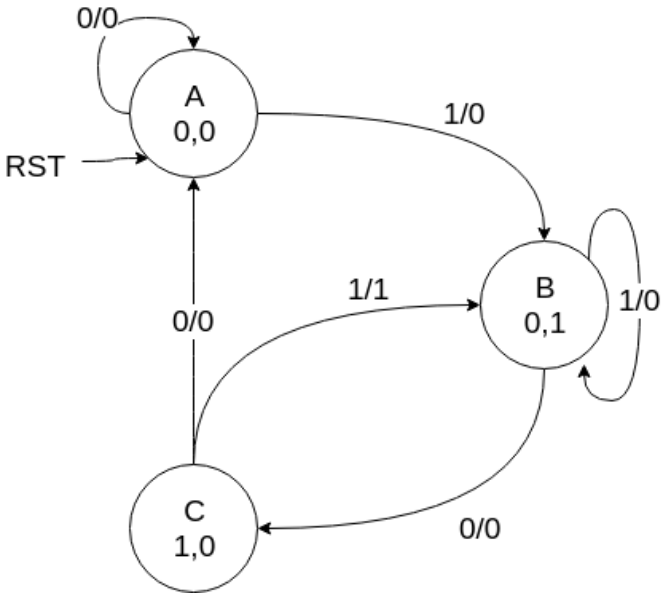


Figure 2: State machine for "101" detector

Table 2: States to binary for "101" recognizer

| State | Binary Representation |
|---|---|
| A | "00" |
| B | "01" |
| C | "10" |

Table 3: Truth Table for State and Output of "101" detector

| Q | A | $Q_0$ | $Q_1$ | $Q_0^*$ | $Q_1^*$ | Output |
|---|---|---|---|---|---|---|
| A | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 1 | 1 | 0 | 0 |
| C | 0 | 1 | 0 | 0 | 0 | 0 |
| n/a | 0 | 1 | 1 | x | x | x |
| A | 1 | 0 | 0 | 0 | 1 | 0 |
| B | 1 | 0 | 1 | 0 | 1 | 0 |
| C | 1 | 1 | 0 | 0 | 1 | 1 |
| n/a | 1 | 1 | 1 | x | x | x |

$$\begin{aligned} Q_0^* &= \overline{x} \wedge \overline{Q_0} \\ Q_1^* &= x \\ Output &= x \wedge Q_0. \end{aligned} \tag{1}$$

Figure 3: Equations for State and Output of "101" detector

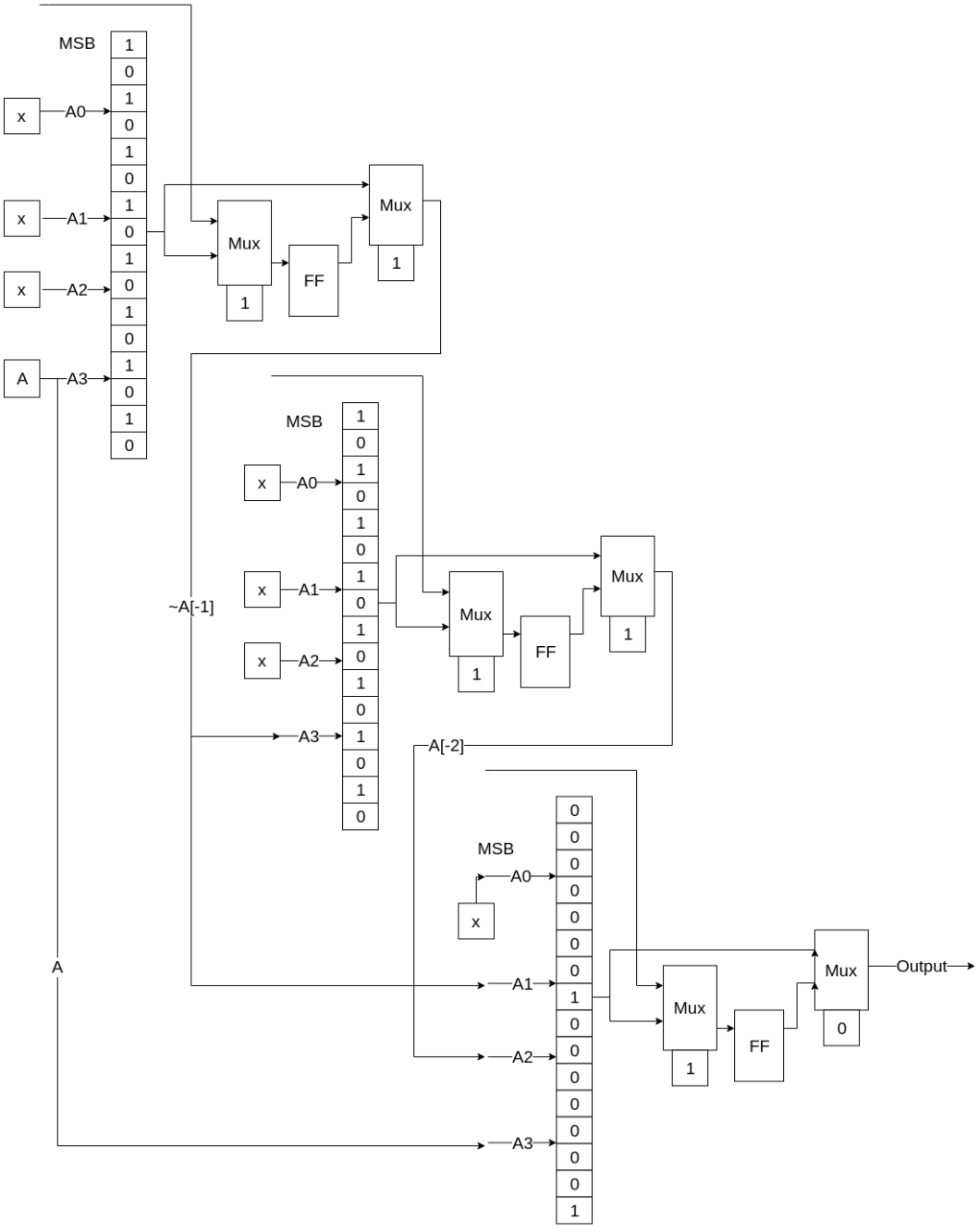


Figure 4: "101" recognizer implemented on simplified logic slices