|  |  |
| --- | --- |
|  | **KHOA CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN MẠNG MÁY TÍNH VÀ TT DỮ LIỆU** |

**Hướng dẫn thực hiện LAB 2.1 - Thực hiện mạch cộng 4-bit**

|  |  |
| --- | --- |
| **Họ tên SV:** | **MSSV:** |
| **Ngày thực hiện:** | **Ghi chú:** |

**Mục tiêu:** Thực hiện 1 IC cộng 4 bit (4-bit adder) và sử dụng tạo thành mạch cộng 32 bits

**Bài học trước:** LAB 1.1 và LAB 1.2

**Thời lượng đề nghị:** 45 phút.

Một bộ cộng đầy đủ (Full Adder) thêm các số nhị phân và tính các giá trị được chuyển vào cũng như chuyển ra. Bộ cộng đầy đủ một bit thêm ba số một bit, thường được viết là A, B và Cin; A và B là các toán hạng, và Cin là “carry in” (giá trị nhớ từ phép cộng của hàng trọng số thấp hơn trước đó, ví dụ 13 + 28 khi cộng hàng chục sẽ là 1+2+nhớ1). Bộ cộng đầy đủ thường là một thành phần trong một chuỗi các bộ cộng, chúng thêm các số nhị phân 8, 16, 32, v.v.

Trong ký hiệu dưới đây, A [4] biểu thị rằng đầu vào được đặt tên là "A" và rộng 4 bit, chứ không phải là một đầu vào có tên là “A [4]”. Một cách tiếp cận sai khác là tạo bốn đầu vào riêng biệt, mỗi đầu vào được đặt tên là A hoặc một số biến thể. Thay vào đó, cách tiếp cận đúng là tạo một đầu vào duy nhất và làm cho nó rộng 4 bit bằng cách thay đổi thuộc tính "Data Bits" của đầu vào. Các thuộc tính xuất hiện ở ngăn dưới bên trái của cửa sổ Logisim khi đầu vào được chọn.

|  |  |
| --- | --- |
| Add4: | C = A + B + Cin; Cout = overflow |
| Inputs: | A[4], B[4], Cin |
| Outputs: | S[4], Cout |

Đầu ra C được tính bằng cách thêm A, B và Cin. A, B và C có thể là số âm dạng bù-2. Nếu xảy ra tràn, Cout đầu ra phải được xác nhận. Trong những trường hợp như vậy, đầu ra C phải tương ứng với giá trị được tính nếu tất cả các lỗi tràn đều được bỏ qua.

# 1. Thực hiện mạch cộng 1-bit Full Adder

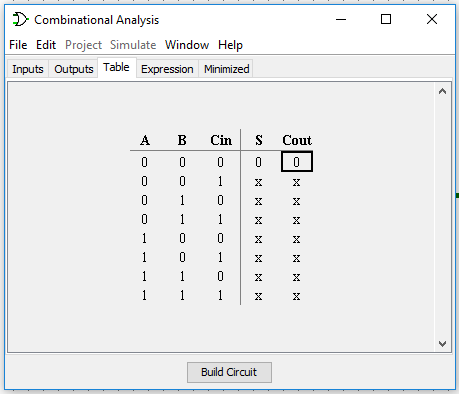
**Yêu cầu 1**: hoàn tất Bảng chân trị sau đây, dòng đầu tiên đã hoàn tất bằng cách tính

S = A+B+Cin = 0+0+0

Cin = 0 do phép cộng S “không tràn”.

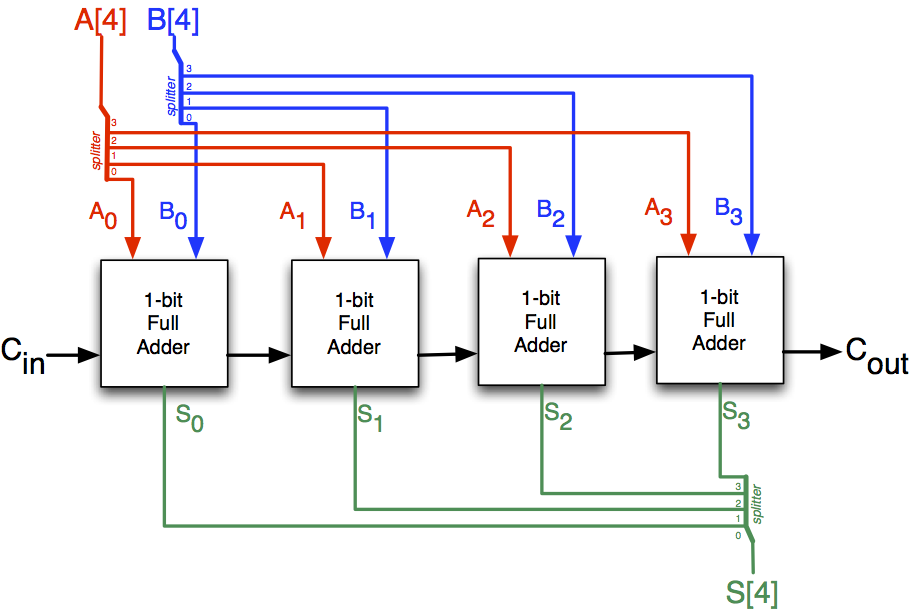
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Inputs** | | | **Outputs** | |
| A | B | Cin | Cout | S |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 |  |  |
| 0 | 1 | 0 |  |  |
| 1 | 1 | 0 |  |  |
| 0 | 0 | 1 |  |  |
| 1 | 0 | 1 |  |  |
| 0 | 1 | 1 |  |  |
| 1 | 1 | 1 |  |  |

**Yêu cầu 2:** Sử dụng chức năng Analyze circuit để hoàn tất Mạch cộng 1 bit dưới dạng mạch con “1-bit adder” bằng cách nhập vào các Inputs, các Outputs và xác lập bảng chân trị rồi dùng chức năng “Build CirCuit”



# 2. Thực hiện mạch cộng 4-bit

Bộ cộng 4 bit đơn giản như việc xếp tầng 4 bộ cộng một bit với nhau, với Carry Out ở mạch này sẽ mang vào Carry In ở mạch tiếp theo. Carry In cho hàng đơn vị sẽ là 0.

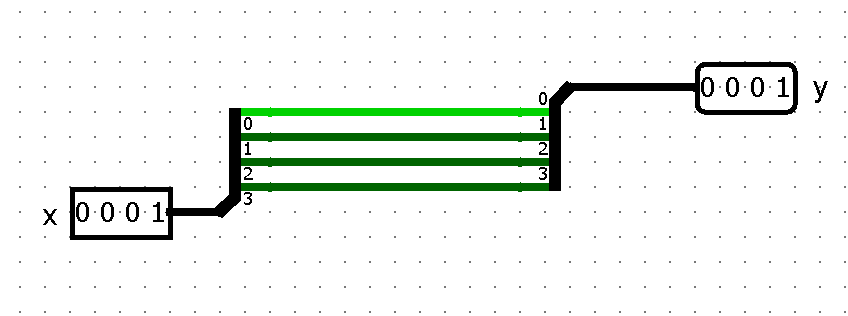


**Hướng dẫn sử dụng Splitter:**

Bộ tách tạo ra sự tương ứng giữa một giá trị nhiều bit và một số tập con riêng biệt của các bit đó. Bất chấp tên gọi của nó, nó có thể chia một giá trị nhiều bit thành các phần thành phần hoặc nó có thể kết hợp các phần thành phần thành một giá trị nhiều bit - hoặc thực sự nó có thể làm cả hai cùng một lúc.

* Facing: Hướng xoay linh kiện.
* Fan-out: số đầu ra tách thành.
* Bus width: độ rộng bus đầu vào.
* Bit x: vị trí tương ứng giữa thứ tự các phần chẻ ra với các bit trong đường tín hiệu. Thông thường sẽ đánh theo thứ tự tương ứng k-k trong đó bit 0 là bit LSB, n là bit MSB.

Dưới đây là một ví dụ của việc truyền 4 bit từ x sang y được Chia ra trước khi truyền và Gộp lại sau khi nhận.

****

**Yêu cầu 3**: Tạo mạch con “**4 bit adder**” trong Logisim, sử dụng lại mạch con là “**Full Adder**”. Nộp tập tin “Lab2\_1.circ” kèm với báo cáo này. Lưu ý các đầu vào của “4 bit adder” đã được mô tả ở đầu bài là A[4], B[4], Cin và đầu ra là S[4], Cout.

**Yêu cầu 4:** Trong mạch Main tạo mạch mô tả việc sử dụng mạch con “**4 bit adder**” này. Ví dụ mạch trừ 4 bit.

# 3. Tài liệu tham khảo

[1] Anne Bracy, *CS 3410: Computer System Organization and Programming*, Cornell University, 2018.

<http://www.cs.cornell.edu/courses/cs3410/2018sp/labs/lab1/>