Building Logic Gates and Adders in Scratch

Introduction

Inspired by the classic educational game *Rocky's Boots*, this tutorial shows how to build logic gates using Scratch sprites and variables as wires. Once you've created the basic gates, you'll be ready to combine them into more complicated digital machines, like adders that can perform math!

Step-by-Step: Building an AND Gate

Sprites and Wires

To start, create three sprites:

- **Switch A**: toggles between ON (1) and OFF (0). Controls variable `A`.
- **Switch B**: toggles between ON (1) and OFF (0). Controls variable `B`.
- **Light Bulb (Output)**: indicates the output of the AND gate.

Then create three wire sprites (thin rectangles) to visualize the signals:

- **Wire A**, **Wire B**, and **Wire AND Out**
- Each wire sprite has two costumes: **gray** (off, signal=0) and **red** (on, signal=1).

Code for the Switch Sprites

Each switch toggles its variable:

```
when sprite clicked
    set [A v] to (1 - (A))
    switch costume to (if A=1 then "on" else "off")
```

Repeat for switch B (variable `B`).

Code for Wire Sprites

Each wire follows its own variable:

```
when green flag clicked
forever
    if <(variable)=1> then
        switch costume to "red"
    else
        switch costume to "gray"
```

Replace `(variable)` with `A`, `B`, or `AND_OUT`.

Code for the AND Gate

The AND gate sprite calculates output:

```
when green flag clicked
forever
    if <(A)=1 and (B)=1> then
        set [AND_OUT v] to 1
    else
        set [AND_OUT v] to 0
```

The bulb sprite shows the `AND_OUT` result.

Creating Other Gates

Create OR, NOT, XOR, NAND, and NOR gates by changing the condition in the gate sprite:

- OR: `if (A=1) or (B=1)`
- NOT: `if (A=0)` (single input)
- XOR: `if ((A=1) and (B=0)) or ((A=0) and (B=1))`

The Half-Adder

A half-adder adds two binary digits (bits) to produce a **sum** and a **carry**.

- **Sum**: result of XOR gate (1 when exactly one input is on)
- **Carry**: result of AND gate (1 when both inputs are on)

Create variables and wires for `SUM` and `CARRY`. Then create a sum LED and carry LED to show outputs.

Scaling Up: Full Adder and Byte Adder

A **full adder** adds three bits: two bits plus a carry-in from the previous bit addition.

How to Build a Full Adder

Combine two half-adders and one OR gate:

1. Inputs: bits `A`, `B`, and `Carry-In`
2. Half-adder #1 adds `A` and `B`, outputs `S1` (sum) and `C1` (carry).
3. Half-adder #2 adds `S1` and `Carry-In`, outputs final `SUM` and another carry `C2`.
4. OR gate combines carries `C1` and `C2`, outputs final `Carry-Out`.

Byte Adder: 8 Bits

To add two bytes (8 bits each):

- Connect 8 full adders in a chain.
- Each bit addition outputs a carry-in for the next bit.

This is exactly how real computers add numbers!

Conclusion and Next Steps

You now know how logic gates combine to build digital circuits like adders. Explore creating bigger machines, build calculators, counters, or even your own mini-computer in Scratch!