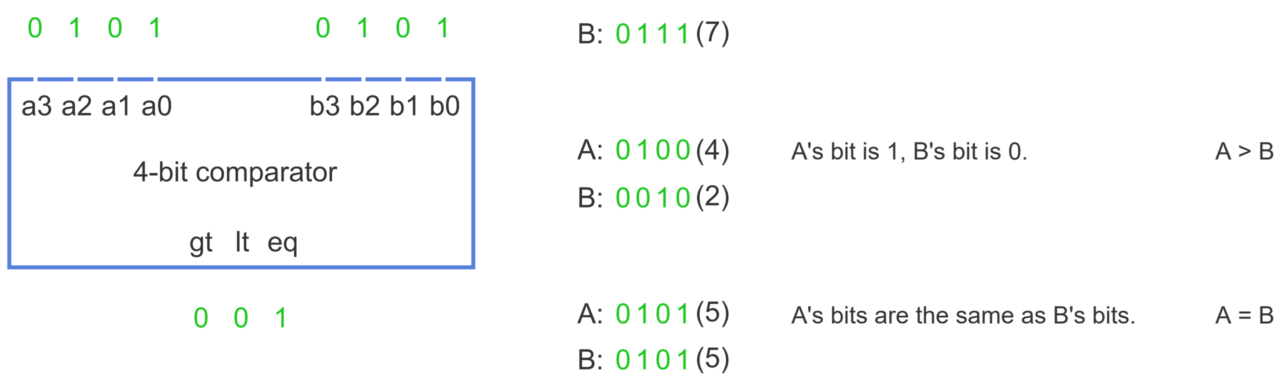
4.3 Comparators

A ***comparator*** compares two numbers, indicating whether the numbers are equal, or which number is greater. Same-length numbers can be compared by hand just like base ten numbers: Starting from the left, digits are compared until a difference

PARTICIPATION ACTIVITY 4.3.1: Comparator.

Start 2x speed

Note: A is a3a2a1a0. B is b3b2b1b0.



PARTICIPATION ACTIVITY 4.3.2: Comparator.

Indicate which comparator output will be 1.

1) A: 1100

B: 1101

- gt
- lt
- eq

2) A: 0100
B: 1000

- gt
- lt
- eq

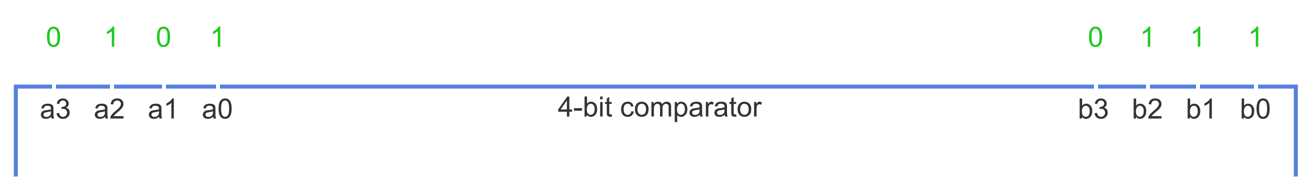
3) A: 1111
B: 1111

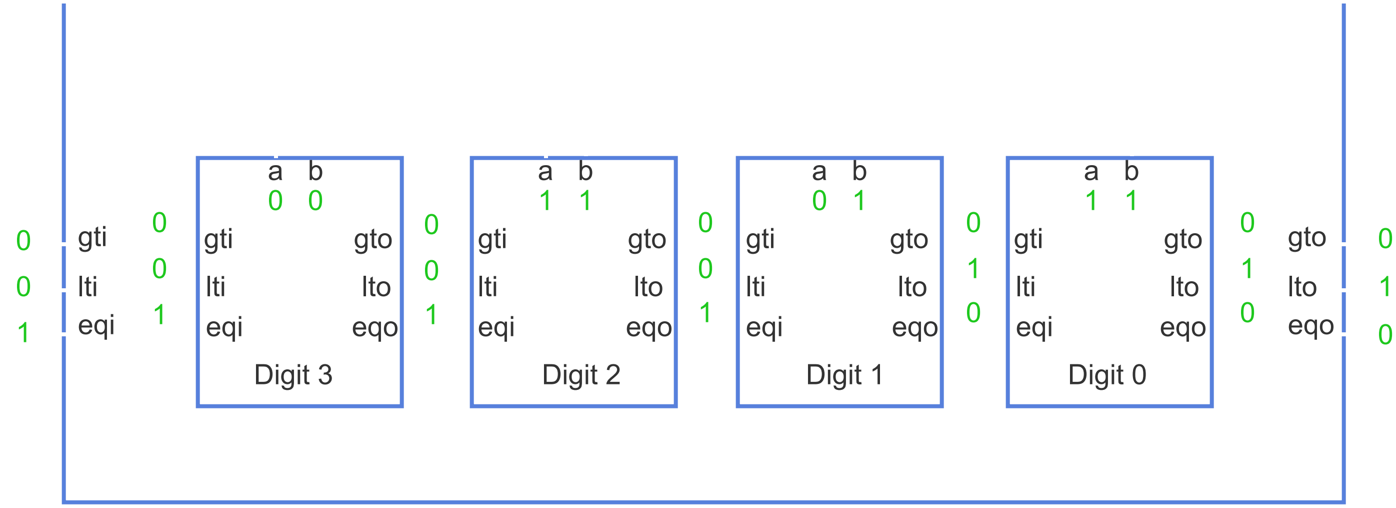
- gt
- lt
- eq

A ***carry-ripple comparator*** compares two N-bit numbers from left to right, with the result of each digit's comparison "ripplin digit. For each digit, a ***one-bit comparator*** compares two bits a and b only if the eq input was 1 from the higher digit, else ju a gt 1 or an lt 1. The rightmost digit's output becomes the N-bit comparator's output. The name "carry-ripple" refers to the s comparator's implementation to a carry-ripple adder's implementation.

PARTICIPATION ACTIVITY 4.3.3: Carry-ripple comparator.

Start 2x speed





The correct output of leftmost digit ripples to next digit, and so on. Eventually the external outputs become correct.

PARTICIPATION ACTIVITY 4.3.4: Carry-ripple comparator.

Consider a 4-bit carry-ripple comparator (seen in the above animation). Indicate which output will be 1 for each digit.

Assume: a3a2a1a0 = 0100 (4), b3b2b1b0 = 0010 (2)

1) Digit 3

- gto
- lto
- eqo

2) Digit 2

- gto
- lto
- eqo

3) Digit 1

- gto
- lto
- eqo

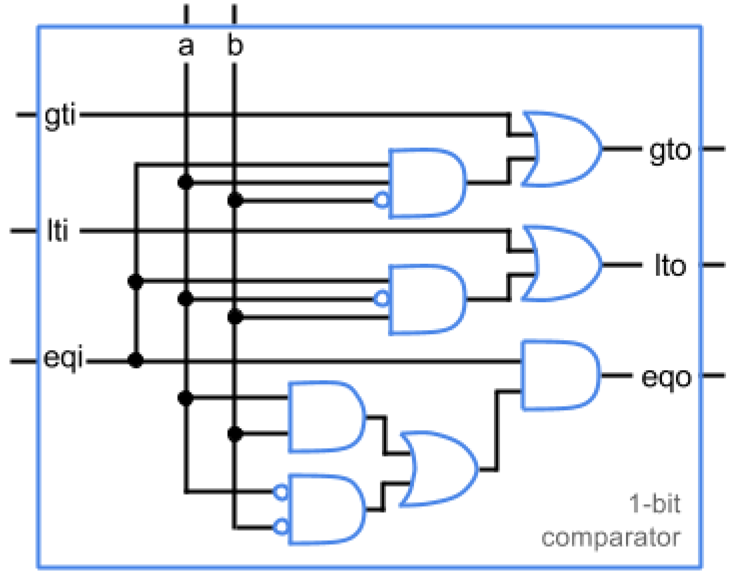
4) Digit 0

- gto
- lto
- eqo

A one-bit comparator can be implemented using combinational logic for each output. A designer could start by filling in a tı inputs gti, lti, eqi, a, and b, and 3 outputs gto, lto, eqo. The truth table will have $2^5 = 32$ rows. Alternatively, the designer can ៜ equations.

- eqo should be 1 if eqi is 1 AND a, b are the same. Thus: eqo = eqi(ab + a'b').
- gto should be 1 if gti is 1, OR eqi is 1 AND ab are 10. Thus: gto = gti + eqi(ab').
- lto should be 1 if lti is 1, OR eqi is 1 AND ab are 01. Thus: lto = lti + eqi(a'b).

Figure 4.3.1: 1-bit comparator circuit (used in questions below).



PARTICIPATION ACTIVITY 4.3.5: 1-bit Comparator.

1) Given: a = 1, b = 1, gti = 0, lti = 0, eqi = 1.

- eqo = 0
- eqo = 1

2) Given: a = 1, b = 0, gti = 0, lti = 0, eqi = 1.

- eqo = 0
- eqo = 1

3) Given: a = 0, b = 0, gti = 0, lti = 1, eqi = 0.

- eqo = 0
- eqo = 1

4) Given: a = 0, b = 0, gti = 0, lti = 1, eqi = 0.

- lto = 0
- lto = 1

5) Given: a = 0, b = 1, gti = 0, lti = 0, eqi = 1.

- lto = 0
- lto = 1

6) Given: a = 1, b = 0, gti = 0, lti = 0, eqi = 1.

- lto = 0
- lto = 1

7) Given: a = 0, b = 1, gti = 1, lti = 0, eqi = 0.

- gto = 0
- gto = 1

Provide feedback on this section