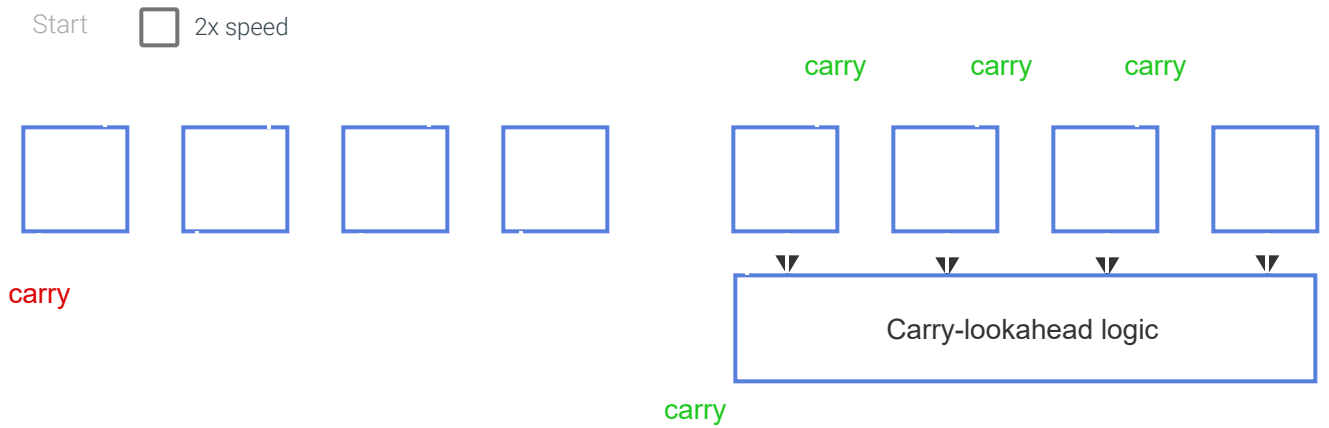
3.28 Carry-lookahead adders

A carry-ripple adder has a drawback of long delay, due to each digit having to wait for carries to ripple through earlier digits. ***lookahead adder*** uses logic to quickly pre-compute the carry for each digit, and thus has less delay than a carry-ripple adde representing a tradeoff.

PARTICIPATION ACTIVITY 3.28.1: Carry-lookahead adder has less delay, but larger size, due to logic that pre-computes carries to avoid rippling.



PARTICIPATION ACTIVITY 3.28.2: Adder tradeoffs.

1) Which adder type has shorter delay?

- Carry-ripple
-

Carry-lookahead

2) Which adder type requires fewer gates?

- Carry-ripple
- Carry-lookahead

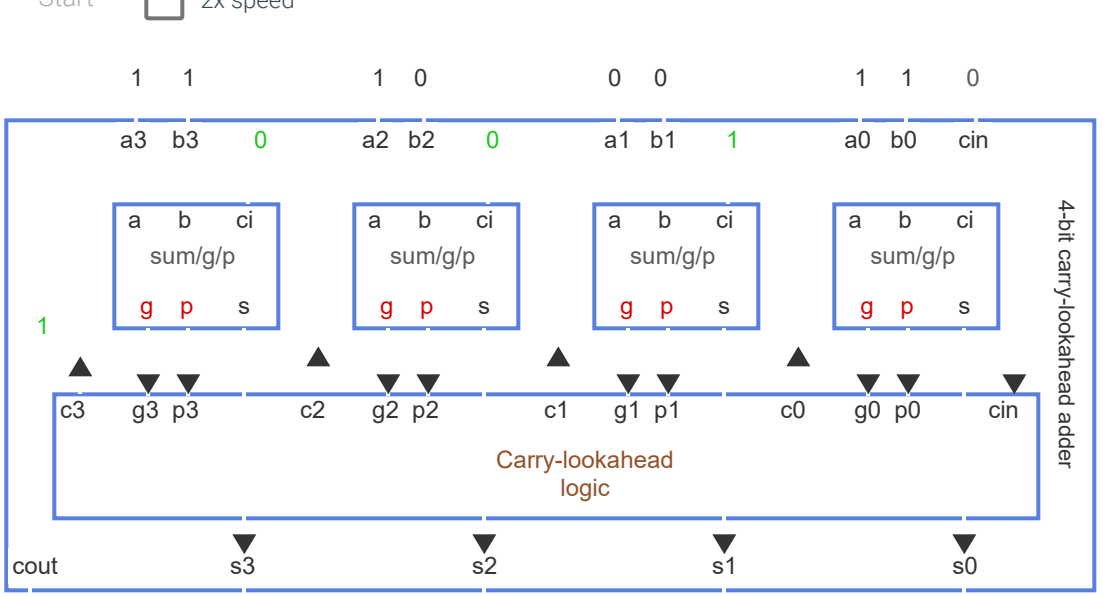
The basic lookahead idea is to create a circuit that quickly computes whether a digit's output carry will be 1 or 0. Each digit circuit makes use of two values, g and p.

- When a AND b are both 1's, the digit ***generates*** a carry-out of 1 regardless of the carry-in. Let g = ab.
- When a OR b is 1, the digit sets carry-out to 1 if the carry-in is 1, akin to the digit ***propagating*** the carry-in to carry-out.

With those two values, the expression for each digit's carry-out is co = ab + (a + b)ci = g + p·ci. Each digit's carry-in can ther with the digit-to-the-right's carry-out (being connected), yielding two-level logic for each carry-in bit. Thus, all carry bits can l without any rippling, at the expense of the large number of gates in the carry-lookahead logic.

PARTICIPATION ACTIVITY 3.28.3: Carry-lookahead adders.

Start 2x speed



a3 a2 a1 a0
+ b3 b2 b1 b0
cout s3 s2 s1 s0

1 0 0 1 0
1 1 0 1
+ 1 0 0 1
1 0 1 1 0

Digit carry-out is 1 if:

ab + (a + b)ci

Let g = ab *Let p = (a + b)*

1 0 1 1 0

c3 = g3 + p3c2 c2 = g2 + p2c1 c1 = g1 + p1c0 c0 = a0b0 + (a0 + b0)cin

c0 = g0 + p0cin

... ...

c1 = g1 + p1(g0 + p0cin)

c1 = g1 + p1g0 + p1p0cin

c2 = g2 + p2g1 + p2p1g0 + p2p1p0cin

c3 = g3 + p3g2 + p3p2g1 + p3p2p1g0 + p3p2p1p0cin

PARTICIPATION ACTIVITY 3.28.4: Carry-lookahead adder.

Consider a 4-bit carry-lookahead adder.

1) Given a0 = 0 and b0 = 1, g0 = ?

Check Show answer

2) Given a0 = 0 and b0 = 1, p0 = ?

Check Show answer

3) Given g0 = 0, p0 = 1, and cin = 0, c0 = ?

Check Show answer

4) Given g1 = 0, p1 = 1, g0 = 0, p0 = 1, and cin = 0, what is c1?
Note: c1 = g1 + p1c0 = g1 + p1g0 + p1p0cin.

Check Show answer

PARTICIPATION ACTIVITY 3.28.5: Carry lookahead size and delay.

Consider a 4-bit carry-lookahead adder. (Note: Assume every gate input requires 2 transistors and ignore inverters.)

1) c0's circuit has ___ gate-delays from the lookahead block's p/g inputs to the c0 output.
   - ○ 2
   - ○ 4

2) c3's circuits has ___ gate-delays from the lookahead block's p/g inputs to the c3 output.
   - ○ 2
   - ○ 8

3) c0's circuit has ___ transistors.
   - ○ 2

- 8

4) c3's circuit has ___ transistors.
   - 8
   - 38

5) Consider 8-bit lookahead logic. c7's circuit would have ___ transistors. Hint: ANDs: 4 + 6 + 8 + ... + 18, OR: 18.
   - 38
   - 106

The lookahead logic gets dramatically larger for wider adders. (And slower too in reality, because for example 18-input gate slower than 2-input gates.) Thus, a designer might construct a 32-bit adder by connecting 8 4-bit carry-lookahead adder blc ripple fashion, for example.

Provide feedback on this section