RIPPLE CARRY ADDER

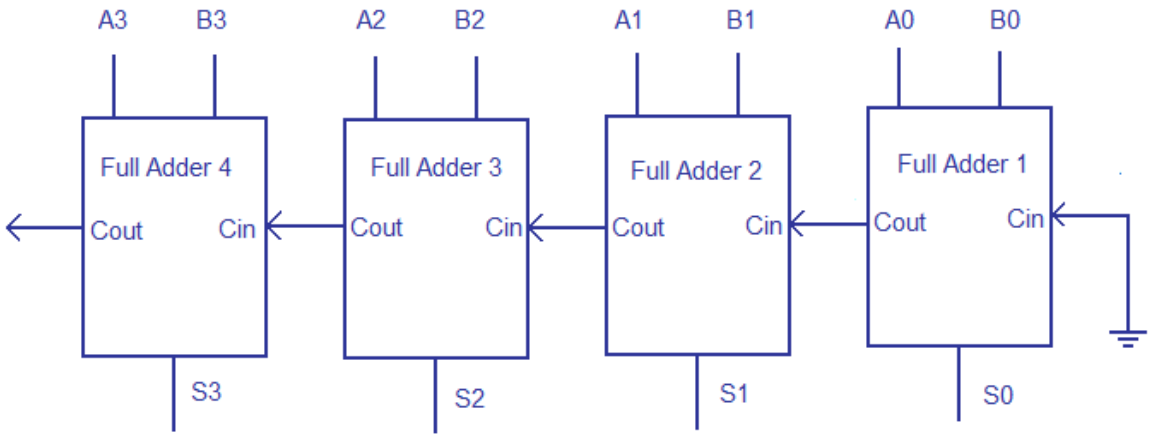
Ripple carry adder circuit.

Inputs: The ripple carry adder takes two binary numbers, A and B, as input. Each bit of the binary numbers is represented as a single line of the input.

Bit-wise Addition: The addition of corresponding bits (same positions) of A and B is performed using simple binary addition rules.
The XOR (exclusive OR) gate is used for the sum bit, and the AND gate is used for the carry-out bit.
Sum Bit (S):=Si =Ai ⊕B i , where Si is the sum bit at position i, and⊕ denotes the XOR operation.

C_{i+1}=(Ai.Bi)+(Ai.Ci)+(Bi.Ci) where C_{i+1} is the carry-out bit at position i+1 and '.' is the AND operation.

Propagation of Carry: The carry-out bit from each bit position is propagated to the carry-in of the next higher bit position. This propagation of carry causes a ripple effect, hence the name "ripple carry adder." Output: The final sum is obtained by concatenating all the sum bits, and the carry-out from the most significant bit (MSB) represents the overflow.



4 bit ripple carry adder

www.circuitstoday.com

Blocking Assignment: In a blocking assignment, the RHS is evaluated, and the assignment is completed before moving on to the next statement.

Non-blocking Assignment: In a non-blocking assignment, the RHS is evaluated, but the assignment does not take effect immediately. All non-blocking assignments in the same time step are scheduled to occur concurrently. The order of execution doesn't matter.

Race Conditions:

If blocking assignments are used in a sequential always block, the simulator will execute statements sequentially, leading to race conditions.

In a sequential circuit, multiple state updates occur simultaneously, especially in cases where state changes depend on the current state. Blocking assignments can introduce dependencies between statements and result in incorrect simulation results.

Latch:

A latch is an asynchronous digital circuit with two stable states (set and reset). It is typically constructed using cross-coupled NAND or NOR gates.

Latches are level-sensitive devices, meaning that their outputs can change as long as the input signal meets certain conditions (level).

Register:

A register is a group of flip-flops used to store multiple bits of information. It can be implemented using a collection of flip-flops and associated logic gates.

Registers are synchronous devices, meaning that they update their outputs only at specific points in time (clock edges). They are used to store data temporarily and synchronize it with the system clock.

Flip-Flop:

A flip-flop is a bistable multivibrator circuit, typically constructed using NAND or NOR gates. It has two distinct states and can be edge-triggered or level-triggered.

Flip-flops are synchronous devices that change state at the rising or falling edge of a clock signal. They are used to store a single bit of data and are the basic building blocks of registers.