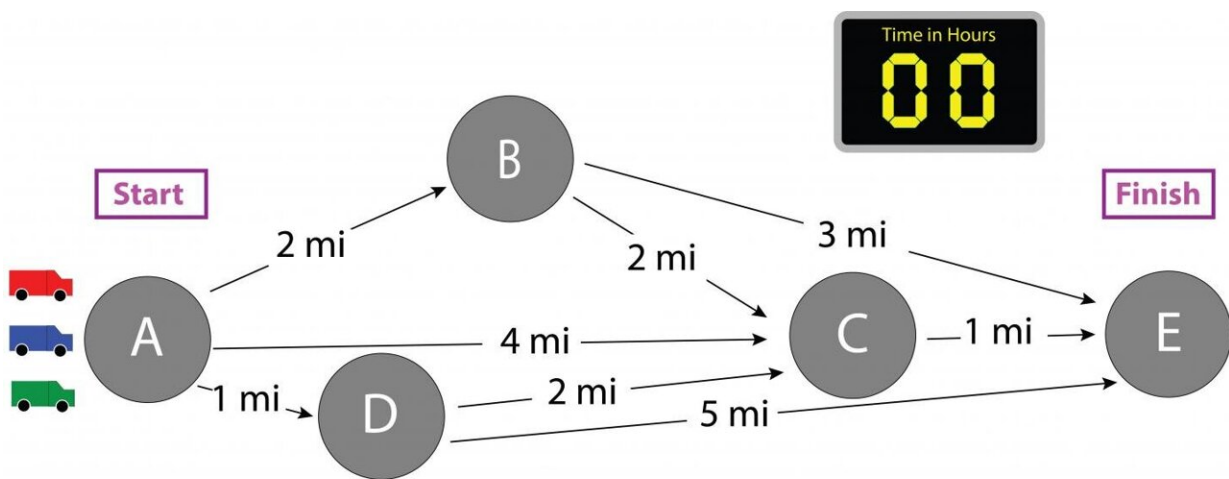# Novel circuitry solves multiple computationally intensive problems with minimum energy

May 11 2021

Numbers next to lines show distance in miles between Cities A, B, C, D, and E.

Each truck travels at the same speed --1 mile per hour -- and each travels in a different initial direction.

The object is to get from the start at City A to the finish at City E in the shortest amount of time.

n/a Credit: NIST

From the branching pattern of leaf veins to the variety of interconnected pathways that spread the coronavirus, nature thrives on networks—grids that link the different components of complex systems. Networks

underlie such real-life problems as determining the most efficient route for a trucking company to deliver life-saving drugs and calculating the smallest number of mutations required to transform one string of DNA into another.

Instead of relying on software to tackle these computationally intensive puzzles, researchers at the National Institute of Standards and Technology (NIST) took an unconventional approach. They created a design for an electronic hardware system that directly replicates the architecture of many types of networks.

The researchers demonstrated that their proposed hardware system, using a computational technique known as race logic, can solve a variety of complex puzzles both rapidly and with a minimum expenditure of energy. Race logic requires less power and solves network problems more rapidly than competing general-purposed computers.

The scientists, who include Advait Madhavan of NIST and the University of Maryland in College Park and Matthew Daniels and Mark Stiles of NIST, describe their work in Volume 17, Issue 3 (May 2021) of the *ACM Journal on Emerging Technologies in Computing Systems*.

A key feature of race logic is that it encodes information differently from a standard computer. Digital information is typically encoded and processed using values of computer bits—a "1" if a logic statement is true and a "0" if it's false. When a bit flips its value, say from 0 to 1, it means that a particular logic operation has been performed in order to solve a mathematical problem.

Time in Hours

02

Start   B   **Finish**

3 mi

2 mi

A   C — 1 mi → E

5 mi

D
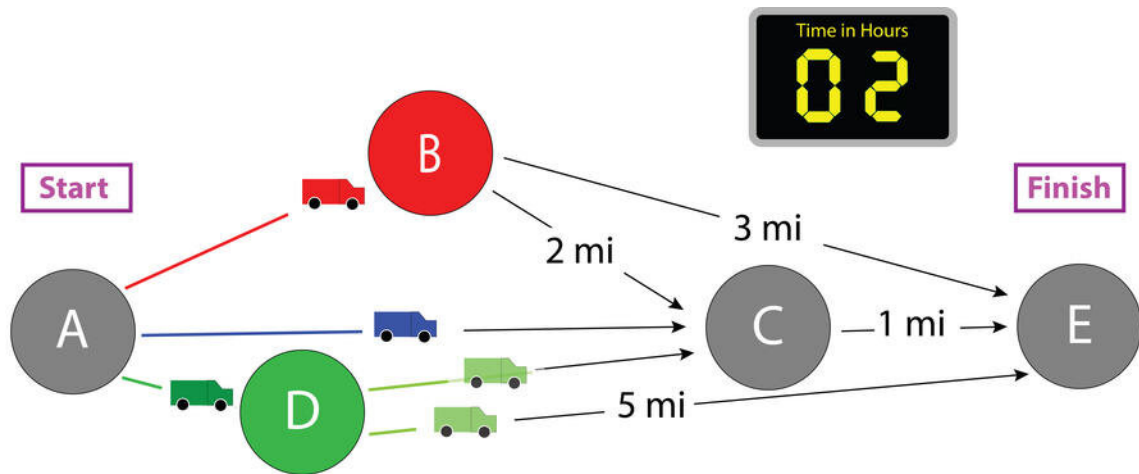
After 2 hours:

The **red** truck reaches City B and claims it for red. The **blue** truck is half of the way to City C.

The **green** truck stops and sends off two related (**light green**) trucks from City D, which it already claimed.

Credit: NIST

In contrast, race logic encodes and processes information by representing it as time signals—the time at which a particular group of computer bits transitions, or flips, from 0 to 1. Large numbers of bit flips are the primary cause of the large power consumption in standard computers. In this respect, race logic offers an advantage because signals encoded in time involve only a few carefully orchestrated bit flips to process information, requiring much less power than signals encoded as 0s or 1s.

Computation is then performed by delaying some time signals relative to others, determined by the physics of the system under study. For example, consider a group of truck drivers who starts at point A and must deliver medicine to point E as fast as possible. Different possible routes go through three intersections—call them B, C and D. To

determine the most efficient route, the race logic circuit evaluates each possible segment of the trip, such as A-B and A-D. If A-B takes more time to travel than A-D, whether it's because the path is longer or has more traffic, A-B will be assigned a longer delay time. In the team's design, the longer time delay is implemented by adding additional resistance to the slower segment.

Race logic does indeed involve a race, but in this contest all the truck drivers initially drive in different directions. To determine which route to the final destination is fastest, they race over all possible routes through the different intermediate delivery points. In the new circuit, the NIST researchers inserted a group of time-encoded signals at the starting point, each acting as a different driver that speeds through the team's simulated hardware circuit.

Whenever a driver arrives at one of her intermediate destination points in the race, the model system sends out new drivers (new time signals) who fan out in different directions to the remaining destinations. If a driver arrives at a destination that another driver has already been to, that driver drops out, because her path is no longer competitive. The winner of the race—the first driver to arrive at the end of the circuit—indicates the solution to the particular puzzle that the hardware was programmed to solve.
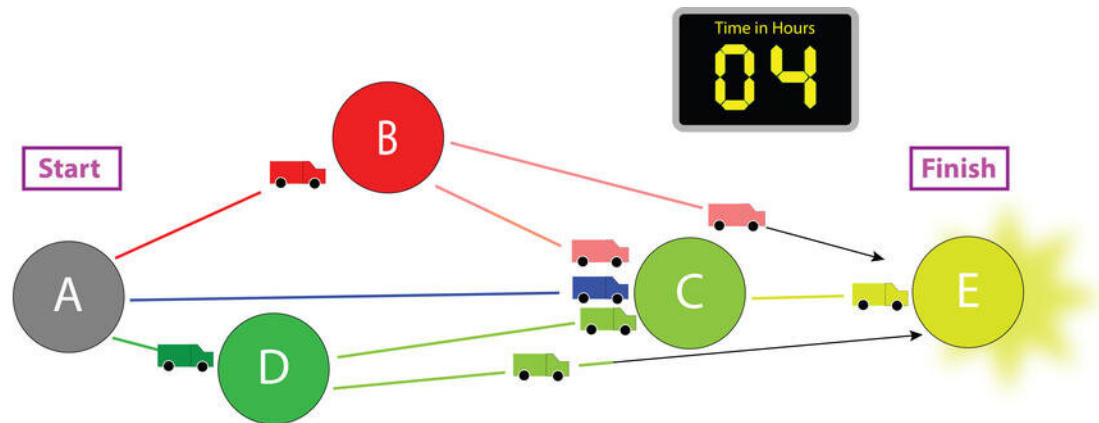
After 4 hours:

Having already claimed City C, the front **light green** truck sends out a **yellow-green** truck, which proceeds to City E and claims it, thus establishing the shortest route from City A to City E: A to D, D to C, and C to E, along all the green-shaded paths. That route is exactly 4 miles long. The **blue** truck and one of the **light red** trucks have arrived at City C, but can't do anything because it has already been claimed.

Credit: NIST

Madhavan began pioneering work on race logic circuits while a graduate student at the University of California, Santa Barbara, in 2016. Those first systems used specialized circuits and silicon chips that were designed to simulate specific networks, such as DNA manipulation, and therefore could solve only a limited number of network-related problems.

At NIST, Madhavan and his colleagues have begun work on more advanced race logic circuits. Simulations conducted by Madhavan, Daniels and Stiles showed that their design, which has not yet been incorporated into a working device, can handle a much broader class of networks, enabling race logic to tackle a wider variety of computational puzzles. These puzzles include finding the best alignment between two proteins or two strings of nucleotides—the molecules that form the

building blocks of DNA—and determining the shortest path between two destinations in a network.

"We showed how to use memory, which has not been used in previous implementations of race logic, to create a more general temporal computer," said Stiles. "Incorporating memory will allow us to treat a broad class of problems with the next race logic chip that we are planning to make," he added.

More information: Advait Madhavan et al, Temporal State Machines: Using Temporal Memory to Stitch Time-based Graph Computations, *ACM Journal on Emerging Technologies in Computing Systems* (2021). DOI: 10.1145/3451214

Provided by National Institute of Standards and Technology