COMPUTER RECREATIONS

An ancient rope-and-pulley computer is unearthed in the jungle of Apraphul

by A. K. Dewdney

On the island of Apraphul off the northwest coast of New Guinea archaeologists have discovered the rotting remnants of an ingenious arrangement of ropes and pulleys thought to be the first working digital computer ever constructed. Chief investigator Robert L. Ripley of Charles Fort College in New York dates the construction to approximately A.D. 850.

The Apraphulians were excellent sailors. Their ships were wonderfully built and equipped with the most elaborate rigging imaginable. Were the Apraphulians led to the digital computer by their mastery of rope or was it the other way around? Experts continue to debate the topic hotly.

The ancient rope-and-pulley computer has recently been partially reconstructed by Ripley and his team at the Tropical Museum of Marine Antiquities in nearby Sumatra. Scouring a site that extends through several kilometers of dense jungle east of the Pulleg Mountains, the group found faint traces of buried jute fibers and noted the exact position of badly corroded brass pulleys and associated hardware. The reconstruction has given me an ideal opportunity to introduce readers to the principles of digital computing without resorting to tiny and mysterious electronic components. Here are gates, flip-flops and circuits made entirely of rope and pulleys. It is all visible and perfectly easy to understand.

The Apraphulians used a binary system just as we do, but the numbers 0 and 1 were represented by the positions of ropes instead of by electric voltages. Imagine a black box with a hole drilled in one side. The reader holds a taut rope that passes through the hole. This position of the rope represents the digit 0. If the reader now pulls on the rope, a creak and squeal inside the box is heard as a foot or so of rope comes out. The new position of the rope represents the digit 1.

One can represent numbers with such boxes. Any number from 0 through 7, for instance, can be represented by three boxes [*see top illustration on opposite page*]. By employing more boxes, larger numbers can be represented. Ten boxes suffice to represent all numbers from 0 through 1,023.

My example of the black box is not arbitrary. The Apraphulians apparently loved to enclose their mechanisms in black wood boxes, small and large. It may be that the construction of computers was the prerogative of a special technological priesthood. The sight of great assemblages of black boxes may have kept the masses trembling in awe.

One of the key devices used by the Apraphulians converted a 0 into a 1 and a 1 into a 0. (It is occasionally convenient to speak of 0 and 1 instead of "in" and "out.") Akin to what modern computer engineers call an inverter, this interesting mechanism consisted of a box with a hole drilled in its front and another in its back [*see bottom illustration on opposite page*]. When someone (or something) pulled the input rope at the front of the box, an equal amount of output rope would be played out of the hole in the back. On peering into the box, the reason is obvious: the ropes entering the box from front and back pass over two fixed pulleys toward one side of the box, where they attach to a single spring.

As some readers may have surmised already, the digits 0 and 1 were not encoded so much by "out" and "in" as they were by the direction in which the rope moved. The point is best illustrated by a box that has no mechanism in it whatever. A piece of rope enters a single hole in the front of the box and leaves by a single hole in the back. If one pulls the rope from the 0 position to the 1 position at the front of the box, the rope moves from "in" to "out." The direction of movement is toward the puller. The rope simultaneously moves from "out" to "in" at the back of the box, but since the direction of movement is still toward the puller, the rope at the back of the box also moves from 0 to 1.

Two additional mechanisms almost complete the ancient Apraphulian repertoire of computing components. The first mechanism had two input ropes entering a box. If either rope was in the 1 position, the single output rope would also be in the 1 position. The Apraphulians managed this trick by absurdly simple means [*see top of illustration on next page*]. Each rope entering the front of the box passed over a pair of pulleys that brought it close to the other rope. The two ropes, passing toward the rear of the box, were then tied to a single ring linked to the output rope. If either or both of the input ropes were pulled, the ring would be pulled directly. Because the output of the box was 1 if one input *or* the other was 1, today's engineers would call this an OR gate.

The ancient Apraphulians fabricated what we would call an AND gate from three pulleys and a curved rod [*see bottom of illustration on next page*]. One of the pulleys was free to roll along the rod, its axle being connected directly to an output rope. The other two pulleys were paired, serving chiefly to position the output rope at the exit hole. With both input ropes in the 0 position, the rod coincided with the arc of a circle centered on the paired exit pulleys. If one of the input ropes was pulled into the 1 position, one end of the rod was pulled away from the center of its resting circle. The pulley attached to the output rope would then roll "downhill" toward the end of the rod that had not been pulled; the position of the output rope would be substantially the same as before since that end of the rod still coincided with the resting circle. (A peg in the middle of the rod kept it from swinging to either side of the box when just one of the input ropes was pulled.)

Only when both input ropes were pulled would the output rope move into the 1 position. In this case the entire rod would have been pulled back into a new position; whichever end the rolling pulley occupied would be equally far from the exit pulleys. The name AND gate is derived from the fact that the output of this device is 1 if and only if one input rope *and* the other are in position 1.

With these components one can build all the control circuits of a digital computer. These include circuits that compute arithmetic functions, interpret program code and direct the flow of information among the parts of the computer.

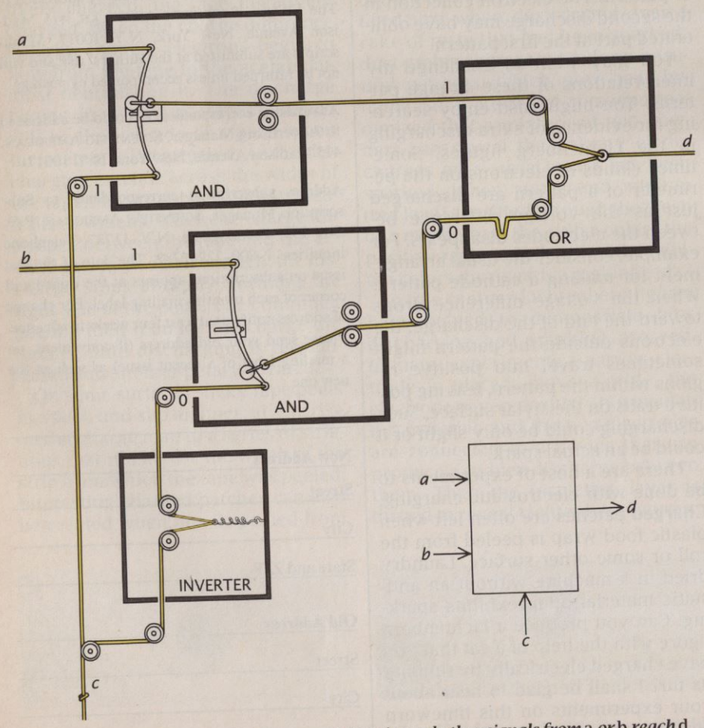
Did the Apraphulians construct their computer along such lines? The evidence is too fragmentary to reach a definitive conclusion, but archaeocomputologists working with Ripley maintain they have discovered a simple multiplexer within the half-buried complex. In electronic computers a multiplexer is essentially an electrical switch that directs the passage of many signals through a single wire. For example, the simplest multiplexer would have two input wires we might label *a* and *b*. At any given moment each wire could carry a 0 or 1 signal. Which of the two signals, *a* or *b*, will be allowed to pass through the device and out a single output wire *d*? The answer to that question is the business of a control wire, *c*; if it carries a 1 signal, the signal from wire *a* will be transmitted along the output wire. If the control wire carries a 0, on the other hand, the signal in wire *b* will be transmitted [*see illustration on opposite page*].

This reconstructed double-input Apraphulian multiplexer consists of two AND gates, an OR gate and an inverter. The whole thing is so simple that one dares to believe computer recreationists might build their own Apraphulian multiplexer at home. Hardware stores might suffer a puzzling run on rope and pulleys. In any event, one can follow operations of the multiplexer by referring to the illustration. Ropes *a* and *b* enter the multiplexer from the top left, each going to its own AND gate. Rope *c* is split. One branch runs directly to the other input port of the AND gate to which rope *a* goes. The second branch of rope *c* passes through an inverter and then runs to the AND gate to which rope *b* goes. If rope *c* is pulled to a value of 1 and held, any sequence of 0's and 1's sent along rope *a* will be faithfully transmitted through the upper AND gate and on to the OR gate. At the same time any signal sent along rope *b* will be stopped at the lower AND gate. If rope *c* is relaxed to its 0 position, the inverter creates a 1 at the lower AND gate. In this case any signal sent along rope *b* will now be transmitted through the lower AND gate and signals on rope *a* will be ignored.

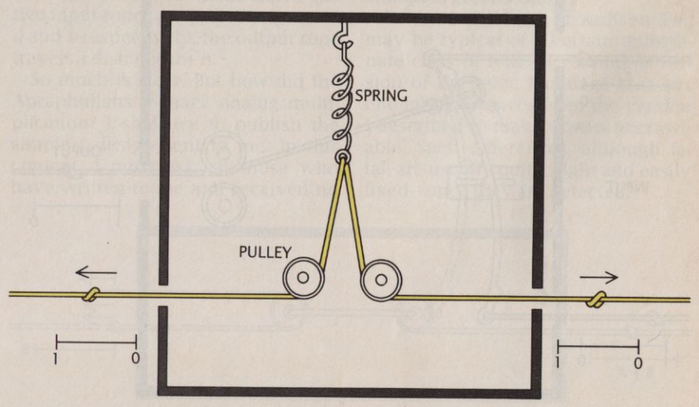
The OR gate merely ties the two

BOX 1	BOX 2	BOX 3	NUMBER
IN	IN	IN	0
IN	IN	OUT	1
IN	OUT	IN	2
IN	OUT	OUT	3
OUT	IN	IN	4
OUT	IN	OUT	5
OUT	OUT	IN	6
OUT	OUT	OUT	7

How Apraphulians represented numbers



An Apraphulian multiplexer: rope c *determines whether signals from* a *or* b *reach* d



The Apraphulian inverter

output signals together, so to speak. If the signal from rope *a* is currently being transmitted, one can easily visualize exactly what happens directly from the diagram: if rope *a* is relaxed to the 0 position, the pulley in the AND box rolls to the end of the rod. A 0 is thus transmitted along the output rope and into the OR box. The other input rope to this box is already in the 0 position (slack). The natural tension on the output rope *d* immediately pulls it into the new position, namely 0. If one pulls on rope *a* again, the pull is transmitted along the path that has just been described, with the result that rope *d* is retracted.

The matter of slack ropes compels me to take up the question of tension in the Apraphulian computer. Sometimes, as in the OR gate of the example, a rope will become slack. There is naturally a danger that such ropes will slip right off their pulleys. Ripley tells me that in such cases the Apraphulians used a specially modified inverter with an extremely weak spring to remedy the problem. Wherever a rope was likely to develop slack, a "weak inverter" was installed to maintain the minimum tension associated with the signal 0.

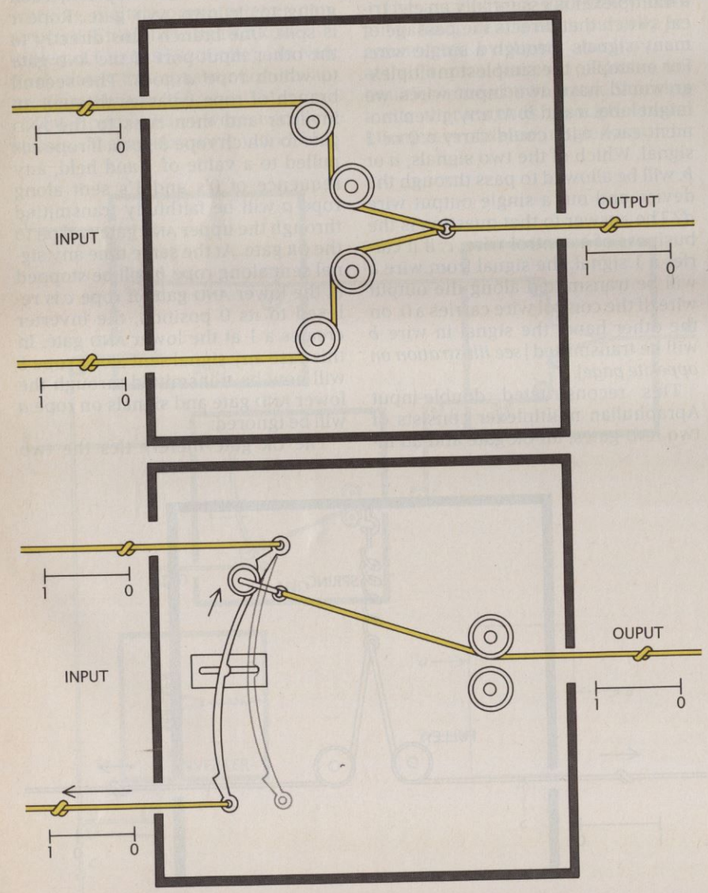
No general-purpose computer is complete without a memory. The memory of the Apraphulian computer consisted of hundreds of special storage elements we would call flip-flops. Here again the remarkable simplicity of the Apraphulian mind is immediately evident. In line with modern terminology, the two ropes entering the mechanical flip-flop are labeled set and reset [*see top illustration on opposite page*]. The two ropes were connected over a series of three pulleys in such a way that when the set rope was pulled away from the box into the 1 position, the reset rope would be pulled toward the box into the 0 position. The common rope was connected to a sliding bar at the back of the flip-flop box. The output rope, physically a continuation of the set rope, had a large bead attached to it that engaged a slot in the sliding bar. As the set rope was pulled, the bead rode over the end of the bar, popping into the slot when the set rope reached the end of its travel.

As a consequence the output rope was held in position until the enormous rope computer changed things by pulling on the reset rope. That had the effect of pulling the sliding bar away from the bead, releasing it and playing the output rope into the 0 position. In this case the flip-flop would henceforth "remember" 0. How were such memory elements used in the Apraphulian computer?

Ripley and his team were puzzled to discover in the midst of the vast Apraphulian computer complex a large overgrown field nearly a kilometer wide. Buried just below the surface of the field were several thousand rotting flip-flop boxes arranged in rows of eight. Ripley, with the aid of the archaeocomputologists, eventually surmised that the field represented the Apraphulian computer's main memory. Each row of eight boxes would have constituted a single, eight-bit "word" in the same sense that the three boxes of my earlier example would have constituted a three-bit word. In that vein, imagine a row of three flip-flops that had been set to the values 1, 0 and 1. They would have stored the number 5.

The content of this particular memory word would have been accessed by the rope-and-pulley computer as follows. Each flip-flop in the row would send an output rope to an associated AND box. The other input to each AND box would come from a special rope used to retrieve the contents of the word in question. When the ropes were pulled, the outputs of the AND boxes would be identical with the outputs of the flip-flops. The AND box ropes would lead to a large assemblage of OR boxes and thence into a special array of flip-flops we would call a register. A single tug on the rope associated with the word under examination would place the same binary pattern of rope positions in the register.

The computer's main logic unit undoubtedly would have directed the flow of information not just from memory to registers but between registers as well. In particular, by



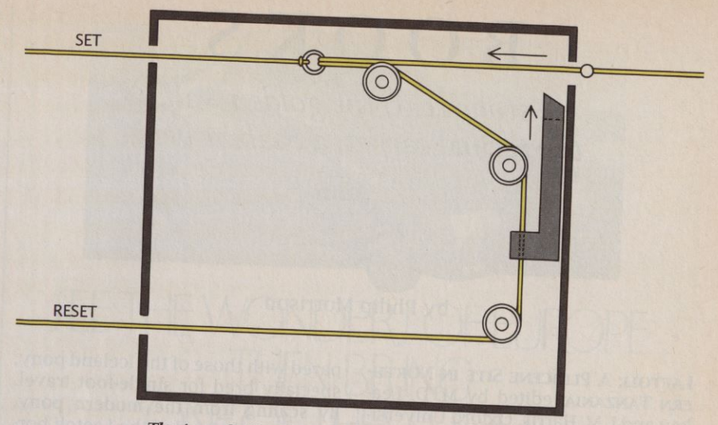
The Apraphulian OR *gate* (top) *and* AND *gate* (bottom)

the use of multiplexers and demultiplexers (which perform the opposite function of multiplexers), the computer would have sent patterns from register to register. At a specific register that we would call the arithmetic register, patterns would have been combined according to the rules of addition and multiplication.

The Apraphulian computer is believed to have been programmable. If it was, part of its vast memory would have been used to store the program. Program instructions would also have been merely patterns of 0's and 1's retrieved by the same mechanism outlined above. Those patterns would in due course have been sent to an instruction register for interpretation by the computer's logic unit.

It is a pity I can do little more in these pages than to hint at the marvelous complexity of the Apraphulian machine. It must have been an amazing sight when in operation. Because of the enormous lengths of rope involved, no human being would have had the strength to pull the input levers into the appropriate positions. The presence of elephant bones in the Apraphulian complex makes the source of input power immediately clear. At the output end large springs maintained appropriate tensions in the system. Perhaps flags on the ultimate output ropes enabled members of the technological priesthood to read the outcome of whatever computation was in progress.

The Apraphulian rope-and-pulley computer makes for an interesting contrast with the nanocomputer introduced in the January column. The rope machine, of course, inhabits a distant past whereas the nanometer-scale machine dwells in a hazy future. The Apraphulian computer is relatively massive in scale, covering thousands of acres; the nanocomputer is incredibly tiny, occupying an area one-thousandth the size of a human cell nucleus. The mere concept of either machine serves as a springboard into a speculative realm where recreation blends with science. Think, for example, of the ongoing dream of artificially intelligent machines. We find it easier to accept the possibility of an electronic computer that thinks since our own thoughts are to a great extent electronically mediated. Because any modern computer (and its program) is conceptually translatable into Apraphulian form, any artificially intelligent device ever realized now or in the future will have its rope-and-



The Apraphulian flip-flop served as a memory element

pulley counterpart. Can we imagine HAL 9000, the paranoid computer in the movie *2001: A Space Odyssey*, being so constructed? Are we willing to admit that an enormous building full of ropes and pulleys could be just as smart as we are?

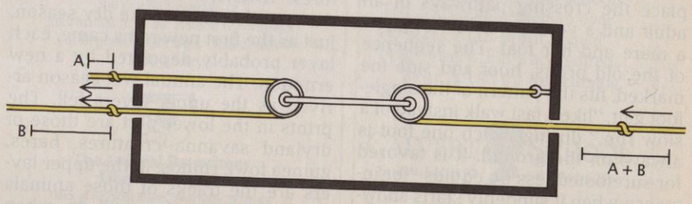
We leave the island of Apraphul with just one backward glance at its misty past: how might the vast digital computer have evolved? From analog ones, of course. The illustration below shows an analog adding machine made from two ropes and two pulleys. The two ends of one rope enter the front of a box through two holes. The rope passes over a single pulley that is linked with another pulley by an axial connector. One end of the second rope is attached to the back of the box. The rope passes over the second pulley and then through a hole in the back of the box. Readers might find some diversion in discovering for themselves how the machine adds two numbers; if the two input ropes are pulled a distance *a* and *b* respectively, the output rope travels a distance $a+b$.

So much is clear. But how did the Apraphulians manage analog multiplication? I shall try to publish the simplest design sent to me. In this context, I must ask all those who have written to me and received no reply to be patient. I am still able to read mail but the large volume prevents me from sending replies to all correspondence.

Most programmers (tyros or otherwise) attempting the special effects described in the December column settled on worms as the effect of choice. Sydney N. Afriat of Ottawa, who was last heard from in connection with the Tower of Hanoi problem in the November 1984 column, notes that the worms are particularly exciting to watch when the program is run in compiled form.

Warner Clements of Beverly Hills, Calif., disliked the wraparound dismemberment of his worms. Rather than allowing the creatures to crawl off one edge and to reappear at the opposite edge, Clements' program alters the direction of motion by +.25 or −.25. The choice depends on whether or not the variable called *change* is greater than .5.

Robert D. Scott, Jr., of Madison, Va., may be typical of a certain unfortunate class of readers running a version of the BASIC language that has enough differences from the version I described to make WORMS uncrawlable. Such differences, although fatal, are usually quite slight and easily fixed—once they are detected.



An Apraphulian adding machine