

AndOrBit – a Tool for Learning Programming Metaphors
Lee Felsenstein
7 Jan. 2016

In teaching programming the first and most significant obstacle is the lack of easily-understood metaphors to explain the internal functions of the computer. Memory is like a set of pigeonholes with addresses – containing numbers (presumably in envelopes). These numbers can be copied out of and written into these envelopes, somehow. A number can be a collection of bits (what are they?), a printable character (or two or four), a “Boolean” that can be “True” or “False”, an integer, a floating-point number, etc., etc.

Small wonder that computer instructors prefer to stay away from this task as long as possible, preferring to steer the student as soon as possible to “Hello World” interactions and to avoid hardware-based issues as long as possible. It is understandable when one hears of how bit operations are increasingly being removed altogether from CS curricula.

This is not serving the future well at all. Teaching computer programming with high-level languages only is not teaching computer science – it is teaching magic, whereby one “says” commands and effects somehow happen as a result.

As a hardware design engineer whose electronics self-education began at age 11 I decided to look to my roots for a solution, creating a hardware tool that enables learners in pre-adolescent years to learn the binary fundamentals at a very deep, visual-tactile fashion.

Recalling my first encounter with the newly-introduced (1978) PAL series of programmable logic chips I remember thinking how these devices would make great learning tools. They were simple and accessible as well as powerful. It was easy to see how they worked and you had complete control over their configuration.

Harkening back to the beginning of computers, the PAL family could be used to create tiny “state machines” - elemental building blocks of computers, invented and defined by none other than Alan Turing, inventor of electronic computing itself. State machines can be created either in hardware or software, and when you can define the same thing two different ways you have a real learning tool.

This is why I say that experimenting with state machines “allows you to see how software emerges from hardware”. PALs worked by having logic equations created by selecting crosspoints on a matrix – inputs as columns, outputs as rows. Wherever there was a connection the input was added as a term in an AND function (all of the selected column signals had to be active in order for the row output to be active). Eight rows were combined through an OR function – this creates a structure known as an “AND-OR tree”, which allows the expression of a wide range of logic combinations.

The trouble was that you never got to see this matrix, and the connections were made by a special programming device that contained a small computer itself. You specified the matrix programming through software, typing equations on a keyboard. It helped a lot to keep a drawing of the matrix and pencil in the connection points as you went – the chip manufacturer even printed such matrix diagrams and their connections with the chip specifications.

So, when considering how to make this structure accessible in a literal “hands-on” fashion to middle-school-aged learners, I came upon an old and thoroughly obsolete technology that would work fine. Called “diode logic”, it was used in a few computers in the 1959 time frame. It became a worthwhile technology to miniaturize into the new integrated circuits that emerged a few years later, at which point no one was interested in remaining with the old way, and for good reason.

Diode logic had drawbacks, primarily relatively low speed of operation as computer speeds continuously escalated. For a learner's device this would be no obstacle. Size was another drawback – for a single interconnection the diodes of 1959 were perhaps 1/2” (12.5mm) long and 1/4” (6.3mm) in diameter. Today, however, diodes have shrunk in size and cost, so that Mini-MELF type signal diodes are too small to be held by the fingers and must be handled with tweezers. They cost a fraction of a cent in large quantities.

The possibility of logic equations brought into being by placing tiny, inexpensive diodes by hand using tweezers was an exact fit for such a learning device, and I began working on a design in the late 1990's. By 2010 I was able to build a large, crude demonstrator device using wires and plugs which enabled me to experiment with young people at the “Hack the Future” events in which kids from 10 to 14 participated. This allowed me to learn how and what the kids could learn and verified that a product would find users.

It became clear that the “courseware” - the instructional material – was crucial and that creating it required the work of many people. This called for an open-architecture process in which a first artifact is created and made available to copy so that people with the appropriate experience and skills would be drawn to it and create learning projects.

The parallel with personal computers, with which I had had much experience, was nearly exact. The best minds in the computer field could not conceptualize how to make a viable product out of proposed personal computers. The first product was a desperate attempt to save a company from failure, and it found a market among a large number of amateurs and semi-qualified professionals who engaged in a collective process of mutual assistance to build a market literally from nothing. The utility of such machines, beginning in 1975, was not evident until the release of the first “killer app” - the spreadsheet – in 1979.

It thus becomes a challenge to replicate such a process. The current ferment of activity in STEM provides an important source of motivation and individual capital – in the form of dedicated time and effort. In 1975 many people had used computers directly through timesharing and wanted to do the same on computers they could individually own. It is my hope that the promise of this state machine device, with connectivity to external peripherals and its ability to be connected together, will capture the imagination of enough STEM practitioners and explorers so that they will begin to communicate and assist each other in building the pieces required for a sustainable industry segment.

This will require not only a good product, but the means to encounter each other, exchange information, and develop cooperation. Computer clubs and publications performed that function in the pre-internet days, but today a website dedicated to the product is necessary. I have set one up which can serve as such an information exchange, and I hope to see more set up by others..

Drawing again on my own experience learning electronics as a pre-teenager, I can now appreciate the

importance of mentoring, which I generally denied myself to my detriment. Complex digital chips require diagnostic ports so that they can be examined from inside so as to discover failures. Since the device here is defined by learners the likelihood of wrong connections is almost guaranteed, and the anguish of trying to figure how to diagnose it would have turned me away from any such effort.

It was therefore necessary to include in the design a diagnostic port that could be made accessible not simply to the owner, but through the Internet to anonymous mentors who can provide the function of cross-generational assistance. This proved relatively easy to do, and a special “Over-the-Shoulder” port allows an external device plugged into the board to translate between a USB port (and thereby the Internet through an external computer) and the elements of the board.

I have not yet designed this external device, but with luck someone else will get there first – I encourage that process – and yet others will have to organize the structure through which mentors are recruited, trained, qualified and made available to learners in trouble.

I am asked by people to whom I describe the device and the system whether I could make a success of marketing it to educational institutions. I respond that while I will not be in charge of that decision, I do not expect to market to institutions. The ideal buyer will be the parent or associations of parents whose children are willing to use the devices in a learning environment. Selling to educational institutions is extremely difficult and faces many obstacles that do not exist in the individual market.

The cost of the device (with its complement of eight daughter boards) is in the several hundred dollar range by my projections (a demonstration of the economic advantages of device integration). Used as a learning device its utility will be limited – it will be cast aside when the learner moves on. Therefore, it will be important to establish a resale marketplace for used systems. Not many companies with new products have any interest in facilitating resale, but the goal here is not so much the growth of one company as the growth of the industry segment.

As to sources of capital, I have already described the human capital required, which I believe to be available. Experience teaches us that this capital will not mobilize itself until a real product becomes available with a fixed price and delivery time. Finding the investment to create this first product would be a daunting task were it not for the recent development of crowdfunding.

We will soon be announcing a crowdfunding effort – in effect, taking advance orders for different forms of the product or tokens connected to the product, with product delivery not guaranteed and the payment considered a donation. The first pilot run of ten units, for example, will be delivered to their donors only after a period of initial use in starting the first stages of the industry segment. They will therefore have a unique history and are intended to be collectors' items. Even with deferred delivery they should fetch a premium price. For lesser donors whose contributions do not reach the level of the product price we will offer tokens including actual PAL chips from a stock of several hundred on hand.

It is my hope and expectation that in a few years it will be considered normal for a middle-school-age student to learn logic design using this product, building simple robotic devices and working with other learners to build larger systems. These learners will then be able to advance to learning programming from machine language to the highest levels of sophistication, given their foundation – all this while in grades 9 – 12.

The concept of waiting until upper-division university to begin instruction in computer science will be seen as absurd. University-level CS students will be expected to enter with significantly greater skills and experience than at present. Those who have not had the prerequisite experience building state machines and larger systems will have to be referred to trade schools and community colleges for remedial work.

I have filed for a patent on this design not in order to maintain a monopoly (the design will be licensed free in most cases) but to prevent anyone else from capturing the design as their property. This has been made necessary by the recent change in patent law to “first to file” from “first to invent”.

I invite anyone with an interest in growing this industry segment and in exploring the opportunities we expect to be created to look into the process we are trying to support. Our website URL is www.andorbit.com – please stop by to browse.