Difference Engine

by Andrew Carol

Building A Calculating Machine Using LEGO® Pieces

Carol's Difference Engine 2. (click for larger image)

History

Before the day of computers and pocket calculators all mathematics was done by hand. Great effort was expended to compose trigonometric and logarithmic tables for navigation, scientific investigation, and engineering purposes. The larger efforts involved rooms of semi skilled people, called 'computers', capable of doing reliable arithmetic who would be under the direction of a skilled mathematician.

In the mid-19th century, people began to design machines to automate this error prone process. Many machines of various designs were eventually built but, the most advanced and famous of these was not. The Babbage Difference Engine.

Because of engineering issues as well as political and personal conflict the Babbage Difference engines construction had to wait until 1991 when the Science Museum in London decided to build the Babbage Difference Engine No.2 for an exhibit on the history of computers.

Babbage's design could evaluate 7th order polynomials to 31 digits of accuracy. I set out to build a working Difference Engine using standard LEGO parts which could compute 2nd or 3rd order polynomials to 3 or 4 digits. I have built two generations of Difference Engines and am designing the third version now.

My first generation Difference Engine could calculate 2nd order differences to 3 digits. Issues with the performance of the 'carry' operation prevented easy expansion to more orders of difference or the addition of more digits. It also had a single "modified" LEGO part in each adder unit which fell short of my goal to use only standard parts.

The second generation machine, documented here, calculates to the same precision, but is capable of being expanded to 3rd order differences and 4 digits accuracy. Other than the paper printed digit labels used to output the answer, standard LEGO pieces are used without modification.

I am currently designing a third generation machine which has changes designed to vastly simplify its construction. This is done with an eye to making it easier for others to build their own Difference Engines using LEGO pieces.

There is a lot more coming to this page over the next few weeks. More details on how it works, and more pictures. Including pictures of the engine broken into its basic adder units as well as close ups of the important mechanisms.

Do you want to contact me about the machine?
I am available at: aecarol@coastside.net ..and... aecarol@woz.org

How Does a Difference Engine Compute?

A Difference Engine uses Newton's method of differences to evaluate a polynomial for successive values of x; 1, 2, 3, etc. Computing the *next* entry in a table can be significantly easier than computing an *arbitrary* entry of the table.

In simple terms the method of differences is based on the observation that if the work has already been done to multiply 5 by 6, that work can be reused to multiply 5 by 7 with the addition of another 5 into the previous total. The multiplication of 5 by 8 likewise can be performed by taking the prior multiplication of 5 by 7 and adding another 5 into that total.

5 × 6 = 30
5 × 7 = 35 adding 5 into the previous value (30)
5 × 8 = 40 adding 5 again, into the previous value (35)

The multiplication of the next successive number is the previous result plus the multiplicand. In this case successive multiplications have been reduced to an identical number of successive additions. As long as we are willing to calculate the table entries in order we can save an enormous amount of work.

This concept can be extended to polynomials. Consider a table of values for the polynomial 2x 2 + 3x + 5,

x	**f(x)** **2x 2 + 3x + 5**	**First Difference**	**Second Difference**

1 10	9	4
2 19	13	4
3 32	17	4
4 49	21	
5 70		

The "f(x)" column represents the evaluation of the polynomial for values of x: 1, 2, 3, etc. The "First Difference" column represents the value which must be added to the "f(x)" value on that line to obtain the following result. 10 plus 9 is 19, 19 plus 13 is 32, etc. The "Second Difference" column represents the value which must be added to the current "First Difference" to find the next "First Difference". Note that for this specific polynomial, each "First Difference" is exactly 4 larger than the previous one.

Using this we can evaluate successive values 2x 2 + 3x + 5, using only two additions for each result. We would add the 2nd difference (4) into the previous 1st difference, then add the newly updated 1st difference into the previous polynomial result to obtain the new result. Repeating this cycle of two additions per result would allow us to evaluate the polynomial as many times as desired.

Any polynomial of order two, that is of the form ax 2 + bx + c, may be evaluated this way. Polynomials of order three, ax 3 + bx 2 + cx + d, can be evaluated by adding a third difference column. This may be extended to polynomials of any degree.

Designing a Difference Engine.

I had been mulling how to build a Difference Engine out of LEGO Technic pieces off and on for several years.

I knew the main operation of a 2nd order Difference Engine would be something like:
Step 1 - Add the 2nd difference into the 1st difference.
Step 2 - Add the 1st difference into the current answer, yielding the next answer.
Step 3 - Repeat from step 1.

A mechanism which would add a first number into a second number would be crucial for the project. I had successfully built several working devices that could add into a second number, but which had an undesired side-effect was the changing of the first number to zero. This is unsuitable for use in a Difference Engine, where numbers must always accumulate in a repeating cycle.

Eventually I found Tim Robinson's very cool "Meccano" Difference Engine #1. He had solved the adder problem with a very clever rotor concept from Babbage's Difference Engine #1 that could passively read one digit and sum it into another digit. Because of the significant differences in parts and technology (plastic versus metal, etc), my Difference Engine shares very little with the Meccano Difference Engine other than the important concept of a rotor based adder.

A few days of work yielded a basic rotor which proved the point that a Difference Engine constructed from

LEGO pieces was at least plausible. I did a lot of experiments with detents and adder layout, at first thinking it would be large monolithic contraption. A single large device proved almost impossible to work with. Simple changes required almost complete disassembly.

I finally decided to explore making the machine out of identical adder units working together. Eventually a standard unit organization started to develop. Each unit would add the digit below into itself, also being able to output its own value to the unit above in a likewise fashion. Carry-in would be on the right side and carry-out would be to the left so that numbers could be read naturally left-to-right. The decision to build the engine out of interchangeable adder units was the single most important factor in the eventual success of the project.

The process was very iterative. The current version of the adder unit would be worked until a particular problem had been solved. The previous version of the adder unit would stand in for an interfacing unit to the left or right, above or below. Once all issues which could be resolved with this version of the adder were dealt with, the previous unit would be disassembled and rebuilt as a clean version of the current adder. What had been the current unit would now be the stand-in unit for debugging the newer unit, and so on in a cycle. This process repeated dozens of times, because as problems were found and resolved new ones became visible.

The most difficult problems consistently proved to be a reliable, smooth power-train as well as the carry propagation. These issues required a great many iterations to resolve. Less difficult was the rotor which went through six basic versions and the detent mechanism which went through only three. Another ongoing issue was the design of a framework structure which would allow a completely modular design that could be broken down into adder units and reassembled in only a few minutes, while being strong enough to support the weight of the complete engine.

General Mechanical Issues.

Plastic gearing and axles are subject to large amounts of flex and gear lash, which can be a significant problem where any level of precision is required. A significant amount of gear lash can be eliminated by using fewest number of gears possible, remembering that larger gears will cover a greater distance with fewer parts.

When practical, the gear train should be in a "high-gear" for most of its length, dividing to the desired ratio as close to the destination as possible. The importance of this for precision movement cannot be overstated. There is a fixed amount of "wiggle" between gears and at a high gear this error becomes a smaller percentage of the total movement. In the case of the adder gear-box dividing first and moving a long distance at a low gear ended up with about ten times the error of moving at a high gear and dividing late.

LEGO gears are optimized for horizontal layout lengthwise along beams with a limited number of efficient gear pairings diagonally or between beams. It can be difficult to ensure an axle in a convenient location rotates in the desired direction at the desired gear-ratio. The main power gear box and digit gear chains underwent literally dozens of redesigns.

Fine adjustments which rely on connection friction to preserve them are ineffective because LEGO parts are plastic and will slip with use. Most adjustments in The Engine are tooth based. Gears can be disengaged, rotated a tooth or more, then reengaged. This has been designed to be easy to do for the digits, and all phases of carry.

I also followed several conventions which simplified construction. Yellow was chosen for the framework because it enhanced contrast of the mechanism and helped showcase internal details. Even length axles are always black, odd lengths always grey, and axles with stops white. Quick release pins are always blue.

Digit Representation.

All digits are represented using a 40 tooth gear with each digit having a four tooth range. Accuracy is maintained with the use of a detent to ensure that each digit snaps into place during rotation. The detent is implemented with a 24 tooth gear configured with a six-way ratchet action. A four tooth rotation of the digit gear (1/10th revolution) rotates the detent gear four teeth (1/6th revolution). The six-way ratchet provides the detent action. The result is a 40 tooth digit gear which clicks in 1/10th rotation increments.

The ratchet tension for the detent is provided by a small shock absorber through a linkage. This was originally done using a rubber band, but the tension varied over time and between rubber bands. A shock absorber with a linkage provides a reproducible amount of tension and is now used for all detents across the entire design.

General Engine Overview.

The Difference Engine is built from three mechanical columns; two adder columns and a carry column.

Each adder column is built from a bottom unit, two stacked adders, and a top unit. Both adder columns are mechanically identical. The carry column is similar in structure to an adder column, being composed of a bottom unit, carry detent unit, full carry unit, and a top unit. The carry units are essentially adder units stripped to only support a carry-in function, allowing the carry column to be narrower than an adder column.

The columns are connected to each other with eleven quick-connect pins. Excepting the quick-connect pins, the only connections between adjacent columns is the meshing of the carry-out gear of a column on the right with the carry-in gear of a column on the left and a chain link drive. When the pins are pulled, each column easily separates from the adjacent column.

Each unit in a column rests securely on the unit below and is held in place with four quick-connect pins. The pins are not load bearing while the machine is operating, but are strong enough to keep the units connected while the machine is lifted from above for transport. Excepting the quick-connect pins, the only connection between stacked units is the coupling of the drive and digit axles through pin based couplers which easily lift apart. When the pins are pulled, each unit cleanly disconnects from the unit below when lifted.

Columns must be separated from each other before they can be broken into basic units. The entire Difference Engine can be broken into basic units in about two minutes. Basic units can be reassembled into columns, and then into a functional machine in about five minutes.

Inserting another adder column would increase precision to four digits Laying another row of adders across all columns would increase the order of difference from 2 to 3, although the machine timing would need to be adjusted.

Rotor and Clutch Adder Mechanism.

The rotor and clutch mechanism is the heart of the adder. It adds the digit of the unit it resides in into the digit of the unit above. Specifically it reads the value represented by the digit-value-pin on the digit-value-axle and adds it into the digit above. The rotor continuously turns in a counter-clockwise direction and is normally running idle (not clutched) three cycles out of four.

During the addition cycle, once every fourth rotation, the engagement-arm will touch the rotor-reader engaging the rotor-clutch and connect the rotor through the rotor-clutch to the digit-output-axle transferring the rotation to the unit above. The rotor will turn the digit-output-axle until the rotor-reader encounters the

digit-value-pin which will release the rotor-clutch restoring the rotor back into idle rotation. The digit-output-axle above will have rotated by an amount equal to the distance the rotor turned before encountering the digit-value-pin. The larger the digit, the further the rotor will turn before encountering the pin, the further the digit-output-axle will rotate before the rotor-clutch is released.

As an example where the digit-value-pin is set to represent the digit "4". The engagement-arm will lock the rotor-clutch as the rotor turns past the "0" position. The digit-output-axle will begin to turn starting the addition of "4" into the digit above. The rotor will turn through the angles representing digits "1", "2", "3", and "4". When the rotor reaches the digit-value-pin at digit position "4" the rotor-clutch will disengage having turned the digit-output-axle through four digits of rotation. The rotor will then resume idle rotation no longer connected to the digit above. The digit "4" has been added into the unit above.

In the case when the digit being added is "0" the rotor-clutch should not engage at all. This is complicated because the engagement-arm is trying to engage the rotor-clutch at the same time the digit-value-pin is trying to disengage the rotor-clutch. This will cause the digit-value-pin to get dragged erroneously turning the digit-value-gear from a "0" to a "1". This is resolved by having the engagement-arm spring loaded so it should "lose" the competition with the digit-value-pin by moving out of the way. In practice the digit-value-pin could still be dragged so during the time when a zero would be read, the digit-value-gear will be locked down by the digit-locking-arm so that it cannot be dragged.

The digit-locking-arm swings into the "locked" position just before a zero would be read, holding the digit-value-gear firmly in place. It then swings back unlocking the digit soon after. It is critical the digit-value-gear not be locked when this digit is being added into by the digit below or when a carry into this digit may occur.

It should be noted that the digit-output-axle above the rotor is rotated 60 degrees counter clockwise from the digit-value-gear below the rotor. This is an artifact of the design of the clutch. It is compensated for by the digit-coupler which connects the digit-output-axle of the lower stage to the digit-value-gear of the stage above.

Digit Display and Vertical Coupling Between Units.

The digit-coupler both displays the current value of a single digit and connects the digit axles of two units vertically. The lower unit's digit-output-axle turns as the digit below is being read and this motion is transferred through the coupler to the digit above adding the two digits together. The upper half of the digit coupler is a "Wheel 20 x 30 Technic" on the upper digit-value-axle, displaying the digits 0 through 9 printed around its circumference. The current value of the upper digit is represented by the forward facing digit displayed on this wheel. A "Technic Wedge Belt Wheel" attached to the lower units digit-output-axle forms the lower half of the coupler.

The upper wheel has five pins arraigned in a circle pointing down, and one empty hole. The lower wheel has one upward facing pin and five empty holes. The pins slide smoothly into the matching holes when the upper adder unit is lowered on to the lower unit. This allows simple coupling and uncoupling. The single empty hole in the top wheel and matching pin in the lower wheel act as a "key" so the units can only be mated in the correct orientation.

When the upper unit is detached and lifted away the pins on the upper wheel mated to the holes in the lower wheel smoothly slide apart.

The alignment between the upper and lower axle of the rotor-clutch mechanism is 60 degrees counter-clockwise. To compensate the upper unit digit-value-axle must be 60 degree clockwise in relation to the lower axle digit-output-axle.

Carry Propagation Between Digits.

Because all of the digits of two numbers are added in parallel, carries must be accounted for during a separate carry-fix-up-stage. When a carry is generated, a flag will be set to be acted on later. After the two numbers are added, every digit with a flag set in the next lower column will be incremented. Since a carry in one digit can trigger a carry in the next digit the carry handling is staggered from least significant digit to the most significant digit. The carry for the 1's column is handled then the 10's, 100's, and so on during the course of a single cycle.

The carry mechanism is based on 3-way symmetry. When a carry is generated, the carry-flag-gear is rotated 1/6th rotation bringing a carry-flag-arm into position so that the carry-propagation-arm can later finish the carry. When the addition-stage is completed the carry-fix-up-stage will start. The carry-propagation-arm will rotate past the carry-flag-gear. If no carry-flag-arm is present, there was no carry and nothing happens. If a carry had been detected, the carry-flag-arm would be pushed by the carry-propagation-arm another 1/6th turn completing the carry. Because of the 3-way symmetry, two 1/6th rotations (totaling 1/3rd rotation) complete the carry and restore the carry logic to its original state ready to detect the next carry.

The carry-effect-gear is slaved to the carry-flag-gear. When a carry is detected and the carry-flag-gear rotates 1/6th turn, the carry-effect-gear also turns 1/6th turn making it ready to perform the actual carry. When the carry-propagation-arm pushes the carry-flag-arm, rotating the carry-flag-gear another 1/6th turn, the slave carry-effect-gear also turns 1/6th turn. This causes a small gear to brush by a detection gear in the next most significant digit incrementing the next digit by '1'. Two rotations of 1/6th total 1/3rd which, because of the 3-way symmetry, restores the carry unit into the normal clear state.

Photos of the Difference Engine.

Front of Engine, Large
Back of Engine, Large
Front Detail of Engine, Large

Acknowledgments.

I would like to thank Steve Wozniak for the generous use of his server space for this page, BrickLink for providing a great place to obtain the rather large amounts of LEGO pieces required to finish this project, and the following Bricklink stores for their excellent service and fast turn around. Without them this project would not have been possible.
BricksLand.com FREE S&H
Precious Princess Palace
Troy's Surplus LEGO - GoB
Sk8r's Bricks
Technic Wreckers
Falling Bricks
BRICK-A-THON
Amazing Bricks