Pro Rege

Volume 42 | Number 1 Article 1

September 2013

Using Design Hierarchy in Digital Logic to Illustrate the Scientific Method as a Human Invention

Douglas DeBoer
Dordt College, douglas.deboer@dordt.edu

Recommended Citation

DeBoer, Douglas (2013) "Using Design Hierarchy in Digital Logic to Illustrate the Scientific Method as a Human Invention," *Pro Rege*: Vol. 42: No. 1, 1 - 8.
Available at: https://digitalcollections.dordt.edu/pro_rege/vol42/iss1/1

Editor's Note: This paper was presented at the Christian Engineering Conference, Atlanta, Georgia, June 2013. It was then published in *Proceedings of the 2013 Christian Engineering Conference*, pp. 99–106. Permission to publish was granted by Steve VanderLeest, Conference Chair.

Using Design Hierarchy in Digital Logic to Illustrate the Scientific Method as a Human Invention

by Douglas De Boer

Introduction

What level of authority should be given to "science?" In a 2008 presidential debate, candidate John McCain said that "[public] policy ought to be based upon sound science."[1] Others say they "believe in global warming" or "evolution" or some other topic of current interest. For the purposes of this paper, *science* is taken as a *body of knowledge* acquired from the outworking of the *scientific method*. The scientific method is defined as the process of systematic observation, measurement, experimentation, and the formulation, testing and modification of hypotheses.[2] These definitions can be reasonably construed to be the type of science and scientific method that John McCain meant and what others who "believe in global warming" or "evolution" or some other topic mean.

The Human Activity of Doing Science and Engineering

Science, as defined above, is a product of human thought and activity. For Christians, the cultural mandate to rule over creation (Genesis 1.28) offers a definition of our human purpose, thus offering direction to our thoughts and activities. The relation of the cultural mandate to engineering is full of complex tensions and ambiguities that make engineering work challenging and rewarding.[3] In our response to the cultural mandate, we might, at times, employ the scientific method. Even though Christians believe that all of our thoughts and activities *ought* to be expressive of our devotion to God's mandates, they are tainted because of our sinful and fallible nature. The Bible informs us that our thoughts and activities are imperfect. Nevertheless, knowledge offered by science is valu-

Dr. Douglas F. De Boer is professor of engineering at Dordt College.

able in its application to everyday life. It is valuable because it offers a degree of prediction of the future or allows us a degree of control over the future or a degree of mitigation of some potential harm in the future. A very important aspect of science is its potential application to utilitarian purposes, which is an aspect of engineering.

Consider in more detail how science and engineering relate to each other. A dictionary definition of engineering defines it as a "branch of science" concerned with the development and modification of "engines" (in various senses meaning machines, structures, complicated systems, etc.) using specialized knowledge, especially scientific knowledge, typically for public or commercial use.[4] In this commonly held view of engineering, *development* and *modification*, which are human activities, are at the true core of the definition; however, the concept of an "engine" is a more memorable part of the definition. In other words, to the extent that the common definitions relate engineering to activities, they tend to highlight the application *of science* (as defined above) as the most important engineering activity, if not the only activity! Furthermore, typical dictionary definitions of engineering tend to draw attention to the products and purposes of engineering rather than to the type of human activities that constitute engineering. It is perhaps no wonder that some people in the general public have vague notions of why one might want to be an engineer. Perhaps some of our students, who may enjoy studying science, have elected to study engineering as an "applied science." Others might be enamored of "engines" (or more typically cars, big buildings, computers, smart phones, robots, artificial prostheses, etc.) and thus choose to study engineering. These are some of the people who could benefit from a more nuanced understanding of science and engineering

Indeed science is an important aspect of engineering, but is it in some way a final authority? It cannot be because engineering is substantially different from scientific activity. Although engineers occasionally do some science, and more frequently they "apply science," it would be inaccurate to characterize all of engineering by that sliver of the whole. Engineering has a goal of *creating* certain types of technological things, not just "engines," whereas science has the goal of *discovering* knowledge. In addition to using scientific knowledge, engineers use mathematical knowledge and knowledge of legal or quasi-legal structures such as various codes (e.g. The International Building Code); they also respond to marketplace information (demand for a new product, warrantee service records, prices of raw materials, etc.); and in general, they employ a wide variety of types of knowledge. The outcome of engineering work is also broader than what the common understanding of "engine" would imply. Engineers participate in government and politics as part of their engineering employment by developing regulations for some types of commerce, for example in patent law, and in standardizing weights and measures and regulating communications technologies such as cable TV, telephone networks, and many other technical services. They perform significant activities not directly related to building and modifying "engines," in the broad sense of the word and not centrally rooted in the application of science, even if in most cases science is not far from the center of the activity.

A text edited by Monsma[5] offers a more complete definition of engineering. It cites five key elements of engineering. First, engineering is a human cultural activity. Second, the activity is done by exercising freedom and responsibility in (obedient or disobedient) response to God's cultural mandate. Third, it involves the forming and reforming of the natural creation. Fourth, it is done with the aid of tools and procedures. And fifth, it is done for practical purposes. Non-Christians may prefer to modify or eliminate the second aspect of this definition. However, this is a key element for Christians. Not only is engineering an activity done in response to God's mandate, but the same can be said for science or, indeed, many other human activities. Recognizing science and engineering as being rooted in human activities, which may or may not be obedient to God's will, is the key element of this definition of engineering that offers an engineer direction toward meaningful engineering work.

It is in responding to God that we human beings find fulfillment and true joy in life. An engineering student who is not aware of this more nuanced definition of engineering could focus on the "engines"—the outcomes of engineering—or focus on only the scientific aspects of engineering and then later become disillusioned with the discipline of engineering on the grounds that engineering, being merely the application of science for utilitarian purposes, offers little enrichment for one's spirit. It is important that students and faculty members in engineering realize the human origins of science (and engineering) so that they knowingly can participate in freedom and responsibility as a response to the cultural mandate, from which joy and fulfillment follow.

Science . . . is a result of human activity, done in obedience or disobedience to God's will.

Teaching Digital Logic

When teaching the subject of digital logic, one finds that the hierarchical nature of the subject offers prime opportunities to instruct students in these fundamental definitions. By discussing the role of hierarchy, one can draw attention to the human activity that originated the body of knowledge used in the design process as well as to the human activity that originated the actual design of a typically complicated digital system. A key concept that needs to be communicated to students is that science, as a body of knowledge, is a result of human activity, done in obedience or disobedience to God's will. This concept helps students distinguish between various conceptions of truth and faith and, in particular, helps students understand that a totally objective kind of truth, free of personal and cultural bias, with nothing taken on faith, is impossible.[6] This point will be illustrated by way of example.

Scientific Abstraction Exemplified

Imagine the lawn of a small neighborhood public park. The park supervisor says, "The lawn needs mowing." If we take this as being either totally true or totally false, then we are treating the statement as a *Boolean statement*. A Boolean statement is a declarative statement that is considered totally true or totally false, regardless of any realistic complexities. We can speak of the *truth-value* of the Boolean statement. The truth-value is either true or false, no shades of grey.

But what if part of the lawn is shady and long and another part is rather sunburned and should not be mowed? If one must work with a single Boolean statement, then one must pick the best fit, either true or false, and act accordingly. Otherwise, one can divide the park into zones, maybe two of them or maybe a million of them, and create a Boolean statement for each zone. Then perhaps a computerized lawnmower could mow just the zones where the corresponding Boolean statement is true. By using more Boolean variables (zones), one can accommodate more of the complexities of a real situation to produce more useful outcomes, probably at increased cost. But even if managing a million zones in the park were practical, that still would not encompass all the reality of the park's lawn. As a matter of utility and economy, we will use as few zones as is practical. We will deliberately omit some or maybe a lot of detail from consideration. This example illustrates the principle of *scientific abstraction*. We are finite beings. We need to focus our attention, which means we must neglect some (hopefully irrelevant) details. That which we include in our consideration and that which we omit from consideration are human choices. In making the choices about what to include or neglect (choosing the number of zones in this example), each person advocating a choice is driven by many factors. One person might place high value on the aesthetics of the lawn, another on the economics of managing the lawn, and still another on the environmental impact of the lawn on the larger surrounding ecosystem, and yet another on some other factor(s). It is a scientific choice only in the sense that this is part of the scientific method of observation and revision of hypotheses. Science itself is not foundational in making these choices.

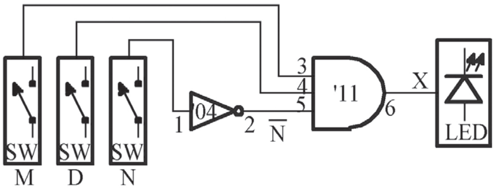
Scientific abstraction, then, is the human process of identifying patterns in real situations by paying attention to only a small but relevant portion of the available detail. What we consider relevant is not, and cannot be, foundationally informed by science; otherwise, we have a circular definition. Abstraction is a type of simplification.

The scientific method then—which was defined in the Introduction as the process of systematic observation, measurement, experimentation, and the formulation, testing and modification of hypotheses—is importantly a process of abstraction, of simplification. Science is merely a process of testing and modifying (improving) hypotheses, which ultimately represents a simplification of a complex situation. Even in choosing and improving the hypotheses, we are exercising our values, a task which is not foundationally a scientific task. The scientific method, applied for utilitarian purposes, is one of many aspects of engineering. (One can point out here that the scientific method is not a process of *proving* hypotheses. Although a hypothesis might be proven false, it cannot be proven true!)

Hierarchy in Digital Logic

The development of a system to detect when the lawn in the park needs mowing might be said to result in a piece of hardware called a "mowing scheduler," to give it a name. Logic gates, which act in accordance with Boolean algebraic mathematical theory, can be used to build the "mowing scheduler." Some logic gates that might be employed are called AND gates, OR gates, NOT gates, and so forth. Each of these will be defined and discussed in a first course on digital logic.

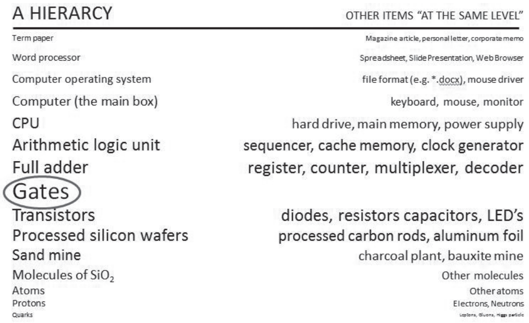
Suppose it is decided that "the lawn needs mowing now" is true only when the lawn is at least four inches tall, and when the lawn is dry, and when it is not nighttime. (Maybe the noise of the mower would bother neighbors at night!) Then a logical circuit could be devised. A design for the "mowing scheduler" is shown in Figure 1. This type of simplistic exercise might be assigned early in a digital logic course.



M = 1 if the lawn is four inches long or longer
D = 1 if the lawn is dry
N = 1 if it is nighttime
X = 1 if "the lawn needs mowing now"

Figure 1. A logic circuit, the heart of the "mowing scheduler"

A simple assignment such as this now affords the opportunity to discuss scientific abstraction and hierarchy. Three levels of hierarchy are easily apparent at this early point in a digital logic course. The concept of the "mowing scheduler" in terms of its inputs, outputs, and desired behavior is itself a level of hierarchy. The person using the "mowing scheduler" will not need to know about how the gates work. The concept of the mowing scheduler as a set of interconnected logic gates (the symbols labeled "04" and "11" in Figure 1) is a second level of hierarchy. The gates are made of interconnected transistors, which are a third level of hierarchy. The engineer designing the "mowing scheduler" and arranging the gates into the circuit does not need to understand the transistors or design the circuit at the transistor level. Observe that one could draw the entire circuit at the transistor level (making no mention of "gates") since, after all, it is just an interconnection of transistors (and possibly a few other parts, also included in the "gates"). This type of hierarchical design is an outworking of the principle of scientific abstraction. To say that the "mowing scheduler" is made up of "logic gates" is a completely human conception. One could just as well say the "mowing scheduler" is made up of various types of molecules in an orderly arrangement, although that description would serve a different purpose.



A HIERARCY	OTHER ITEMS "AT THE SAME LEVEL"
Term paper	Magazine article, personal letter, corporate memo
Word processor	Spreadsheet, Slide Presentation, Web Browser
Computer operating system	file format (e.g. *.docx), mouse driver
Computer (the main box)	keyboard, mouse, monitor
CPU	hard drive, main memory, power supply
Arithmetic logic unit	sequencer, cache memory, clock generator
Full adder	register, counter, multiplexer, decoder
Gates	
Transistors	diodes, resistors capacitors, LED's
Processed silicon wafers	processed carbon rods, aluminum foil
Sand mine	charcoal plant, bauxite mine
Molecules of SiO_2	Other molecules
Atoms	Other atoms
Protons	Electrons, Neutrons
Quarks	Leptons, Gluons, Higgs particle

Figure 2. Example of a Hierarchy

Those who teach digital logic courses will recognize levels in the hierarchy illustrated in Figure 2, such as gates, full adders, arithmetic logic units, and CPUs as possibly being included in their course. These courses typically explicitly include topics at several layers of hierarchy that are all studied in considerable detail, meaning that both the internals (why it works, how to design it) and the externals (what you can do with it and how to use it) are included in the course.

A theory of hierarchies is probably a topic for a paper in itself. For the purposes of this paper, especially as used with digital logic, consider a hierarchy to be an arrangement of humanly defined perspectives (or theories) with some being considered "above" and some "below" and others "at the same level," as exemplified in Figure 2. Objects can manifest themselves at many levels of hierarchy. The highest level at which the object manifests without being subsumed into a larger system is called the *root* level of hierarchy for that physical entity. For example, a gate could be said to be made of transistors, which are made from silicon wafers, which are made from molecules, etcetera. In the conception of the physical object of a gate, the "gate level of hierarchy" is the root level.

Digital logic courses usually start with a presentation of gates at the gate level of hierarchy. "Gates" are presented to students in terms of black boxes that behave in certain prescribed ways to correspond with the rules of Boolean algebra. Usually the internal details (at the transistor level) of gates are not very much discussed, as professors leave this topic for a different course, which in some cases a student might legitimately never study. Thus, hierarchy is a means by which we deal with complexity.

Once the behavior of gates has been presented (and hopefully learned!), the course usually proceeds to discuss how useful things might be built from gates. These useful things cannot themselves be internally understood or designed without reference to lower level(s) of hierarchy—to gates. To be "internally understood" means to be designed (in the case of engineered objects) or to devise theories as to why the objects behave as they do (in the case of non-engineered objects). However, the objects can be used and externally understood without reference to lower levels of the hierarchy. To be "externally understood" means that what may be usefully done with the perspective presented at a particular level of hierarchy, or how to predict behavior at that level of hierarchy, is understood.

As an example of internal and external understanding at a level of hierarchy in the context of a digital logic course, consider a circuit called a *full adder*. For readers not familiar with a full adder, it is a logical circuit that performs addition of two (binary) digits. The inputs to a full adder are each of the two digits to be added, symbolized by *A* and *B,* and also a third input to accommodate the possibility of a carry from the addition of the next lessor significant digit, C_{in}. The outputs are the sum, *S*, and also another output, C_{out} to accommodate the possibility of a carry (out) from the addition.

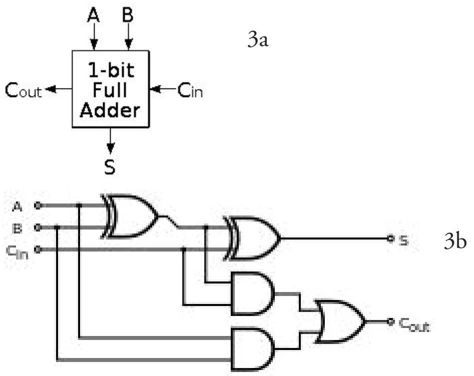


Figure 3. Two illustrations of a full adder[7]

Two illustrations of a full adder are shown in Figure 3. Each symbolic illustration represents the same physical entity but at different levels of hierarchy and for different purposes. Figure 3a shows the inputs and outputs of the full adder, giving emphasis to how to connect the full adder into a larger circuit. Figure 3a is useful in the sense of "how to use" a full adder. This level of hierarchy illustrates our humanly devised external understanding of how a full adder behaves. One cannot understand why it works by making reference to only Figure 3a. This could be labeled as the "full adder" level of hierarchical understanding of this physical entity.

> *What we consider relevant is not, and cannot be, foundationally informed by science; otherwise, we have a circular definition. Abstraction is a type of simplification.*

The entity is also called a full adder. This is the root level of this entity.

Figure 3b shows how to build a full adder by connecting various gates together. To those who possess an external understanding of gates (how they behave), it explains how and why the full adder works. The figure is only useful if one has an external understanding of gates or wants to build a full adder from gates; thus, it gets designated as the "gate" level (of hierarchy) perspective of a full adder.

Textbooks in the field of digital logic go to some length to describe hierarchies in digital logic. Hardware design languages explicitly define syntax with which to encode hierarchies. These structures, which are a normal part of a course on digital logic, should then be examined philosophically, as such an examination allows the professor to point out that each level of hierarchy is a humanly created concept, a scientific abstraction. The authority with which the standard hierarchies are presented in typical textbooks tends to foreclose our thinking of other possibilities, but indeed, the hierarchies we routinely use could be differently arranged and in the past have been differently arranged. For example, the concept of small-, medium-, large-, and very-large-scale digital integrated circuits represents a type of hierarchy that has become passé and is rarely mentioned anymore.

A more interesting example of how hierarchy can foreclose or open our thinking might be to consider the full adder circuit again. How might a full adder be used? Students will naturally expect to connect the carry output of one adder to the carry input of another in order to add numbers having several digits of significance. This could be an example of foreclosure encouraged, once a certain hierarchical idea is embraced. Figure 4 illustrates this connection, called a *ripple-carry adder*. Figure 4a is drawn at the "4-bit adder" level of hierarchy, the root level for this entity. Figure 4b shows how to connect four full adders together to make a "4-bit adder." This is the "full adder" hierarchical level perspective of the entity known as the 4-bit adder. Note that each of the full adders could be re-drawn at the gate level of hierarchy, giving the gate level perspective of the 4-bit adder. The process can be continued to lower and lower levels of hierarchy as far as desired, but the resulting complexity would in most cases cloud our human understanding of how a 4-bit adder works or how to design a 4-bit adder. Usually the root level and one or, at most, two lower levels of hierarchy give the most important external and internal understandings of the entity.

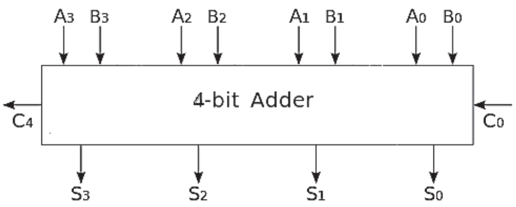


Figure 4a. The ripple carry adder illustrated at the "4-bit adder" level[8]

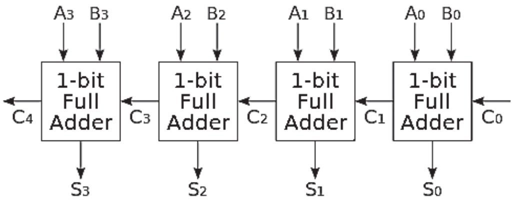


Figure 4b. The ripple carry adder illustrated at the "full adder" level[8]

The 4-bit ripple-carry adder suffers from a long propagation delay because, for example, the S_3 output will not be correctly computed until the carry going into the most significant full adder is computed, a procedure that cannot be done until the preceding carry is done, etc. Correct computations of sum bits will happen first for the least significant bit and then "ripple" to the most significant bit. The choice of hierarchy has encouraged this particular design. This choice may have been motivated by the method of performing addition with pencil and paper, in which correct results also "ripple" from least to most significant.

Other hierarchies are possible for understanding binary addition, and these may offer improved performance. The full adder concept can be replaced with a circuit called a partial full adder (PFA). The PFA has three inputs—*A*, *B*, and C_{in}—defined the same as for a full adder, and three outputs—*S*, *P*, and *G*. The sum, or *S* output, is the same as for a full adder and will not be correct until the carry input is correct, but *P* indicates if the *A* and *B* inputs will cause a carry-in to be propagated across this bit position, and the *G* output will indicate if the *A* and

B inputs will cause a carry to be generated at this bit position. Since the *P* and *G* outputs depend only on the *A* and *B* inputs, the PFA can compute them rapidly without depending on any ripple phenomenon. Ironically, a PFA computes more information (three outputs) with fewer gates. It is a re-conception of hierarchical boundaries.

> *The best choice of hierarchy is simply one that apportions the degrees of abstraction to levels manageable by humans and is simultaneously adequately inclusive of relevant details of the reality of the situation.*

Now another circuit, called a carry look-ahead unit (CLA), rapidly computes all the carries from all the *P* and *G* outputs of the PFAs. Those carries are connected to the carry inputs of the PFAs, and sum outputs get computed much faster than with a ripple-carry adder.

Still other hierarchies exist for binary addition. As an example, possibly useful in some cases, a memory could be used to store a huge addition table. Then the numbers A and B are used as addresses into the table, and the answer is simply looked up in the table.

Hierarchies are a consequence of our human finitude. Even Adam, before the fall, would employ hierarchies. The effects of sin are a burden to correct understanding, but a world free of sin, the new heavens and the new earth, still cannot offer full understanding of the universe to human beings. Real-world digital systems often are so complicated that no one person can fully understand the design and manufacturing techniques needed for all aspects of the system. However, a team of engineers, technologists, and others in proper relationship to God and themselves can work productively with each person assigned to just one or a few of the hierarchical aspects of a system. In this sense, hierarchy is neither to be specifically encouraged, by our devising as many layers of hierarchy as possible, for example, or discouraged, by our minimizing layers, for example. The best choice of hierarchy is simply one that apportions the degrees of abstraction to levels manageable by humans and is simultaneously adequately inclusive of relevant details of the reality of the situation.

In the subject of digital logic, at least, lower levels of hierarchy tend to be best understood primarily in relation to scientific and mathematical theories. Transistors and logic gates can be primarily understood in terms of physics and Boolean algebra, for example. Consider that the quality or fitness to a purpose of a transistor or logic gate is relatively easy to quantify in terms of *numerical* specifications—a datasheet if you will. This is not to say that non-scientific and non-mathematical aspects, such as recyclability for example, never enter into consideration, but that most of the many considerations tend to be scientific and mathematical at the lower levels of hierarchy. As one proceeds in study to higher levels of the hierarchy, the theories needed involve more non-mathematical and non-scientific aspects. For example, what makes a computer processor or a smart-phone good? This is not prominently a question of science or math: the design of a good user interface for a smart phone has elements of aesthetics and psychology. Keeping communications appropriately private has to do with an understanding of law and ethics. These are all elements of the engineering design of a smart phone. Design work done at the higher levels of hierarchy involves different qualities of human judgment and human genius or error compared to work done at lower levels of hierarchy. Christians can understand that choosing the various types of knowledge they use in the design of different levels of hierarchy is part of their response in freedom and responsibility to the cultural mandate.

When a digital logic design is completed and the digital system is finally constructed and put in service, it is a whole piece, not a collection of hierarchical layers. Real-world phenomena will play their roles regardless of the degree to which they were considered during the design work. For example, the effect of a high humidity environment might have been overlooked in the design. Later it might become apparent that corrosion of a particular connection upsets normal operation. Humidity could have been considered, and the designers might have taken steps to mitigate the effect of humidity (or any other real-world phenomenon), but we are finite beings. We cannot perfectly know in advance what can be ignored and what cannot be ignored. Our trust in a design increases to its ultimate level only in response to actual performance in realistic situations. Simulation can also help increase trust, but not as completely as real-world performance. Hierarchies of scientific abstractions are essential to our understanding, yet they also represent our lack of understanding of the totality of the situation.

Conclusion

What level of authority should be given to "science" then? Science, as a body of knowledge, has its most significant roles in the lower levels of hierarchical structures. In that sense it has a foundational role. On the other hand, science is not a final arbiter at any level of hierarchy, and especially so at the higher levels of hierarchy. This limited but important role for science can be discussed in digital logic courses. Christians can consider the various types of knowledge needed for the design of various levels of hierarchy as a response to the cultural mandate. This understanding offers joy and fulfillment in the engineering task. Including this type of discussion, particularly in a course on digital logic, helps students relate what they are learning in the digital logic course to their faith and to other courses they may be taking, even non-engineering courses.

Endnotes

1. "Science Debate 2008," http://www.sciencedebate.org/debate08.html, accessed online 3/25/2013.
2. "Scientific Method," *Oxford English Dictionary*, accessed online 3/25/2013.
3. Kevin Timmer, "Putting the 'And' Back into Genesis 2:15," 2009 Christian Engineering Educator's Conference Proceedings," http://www.calvin.edu/academic/engineering/ces/ceec/2009/CEEC2009proceedings.pdf accessed online 5/20/2013, 3–9. (Revised and reprinted in *Pro Rege*, Vol. XLI, No. 4, June 2013, 27–34. http://www.dordt.edu/publications/pro_rege/crcpi/Pro_Rege_June_2013.pdf accessed online 8/13/2013.)
4. "Engineering," *Oxford English Dictionary*, accessed online 3/25/2013.
5. Stephen V. Monsma, ed., *Responsible technology: a Christian perspective* (Grand Rapids, MI: Eerdmans, 1986), 18–21.
6. David E. Newton, "Science and Religion," in *Social Issues in Science and Technology: an Encyclopedia* (Santa Barbara, California: ABC_CLIO Inc., 1999), 250-253.
7. Illustration from "Adder (electronics)" in *Wikipedia*, http://en.wikipedia.org/wiki/Adder_(electronics), accessed 5/30/2013. For a more academic discussion of a full adder, see M. Morris Mano, Charles R. Kime, *Logic and Computer Design Fundamentals*, 4th ed. (Upper Saddle River, NJ: Pearson Prentice Hall, 2008), 152–153.
8. Ibid.