CHAPTER 2

Fundamentals of Biodesign

In this chapter, we explore the process of *biodesign*: designing biological systems with predictable behavior. We will consider an analogy to illustrate the usefulness of abstraction in design. Then, we'll walk our way through the design abstraction to specify a system for detecting arsenic in water. Along the way, there will be prompts so that you can apply these illustrations and examples to the design of a living technology of your choosing.

Illustrating a Top-Down Design Approach

Biodesign might be new to you, so we'll begin with a more familiar scenario.

Imagine that you want to go on a vacation. First you need to decide what type of vacation. Do you want a tropical getaway, a backpacking trip, an urban adventure? To decide, you might consider the benefits of each of your options. For example, a tropical getaway might be relaxing, but it's also quite expensive. For now, let's say you're feeling more interested in nature than cities, so you decide that the backpacking trip is the best option.

Having decided on the "big picture," a backpacking trip, next you need to begin planning your trip, starting with still more "big" questions: do you want to hike in the mountains or the desert? Do you want to go for two nights or five weeks? For this example, let's say that you decide to go on a three-day trip to the mountains. But now you must decide which mountains you'll visit. Will you choose to hike the Appalachian Trail or Yosemite? If you're living on the West Coast and you have only three days of vacation, Yosemite appears to be the better choice. It seems natural to consider the big picture questions like these as you plan—or design—your trip.

Having decided on a three-day trip to Yosemite, you need to make yet more decisions. Yosemite's a big place. Do you want to stay in the valley or go to the high country? Let's say you know you want incredible views and you don't mind a steep climb or some crowds, so you decide to climb to the top of Half Dome, the iconic peak overlooking the valley.

Having made this choice, you next need to get more specific and decide how you're going to get to Half Dome. A number of different trails can get you there. To plan the route you'll take, you might sit down with a detailed trail map to decide the number of miles you want to hike, your desired elevation change, and the proximity you'd like to stay to water. These details will make a difference in the way you climb Half Dome and how far along you'll get.

Finally, with the trails chosen and the details of your climb in mind, you need to make a list of the equipment you need. Some of these pieces of equipment might be standard, such as hiking boots and sunscreen, but there might also be some special items, such as the rubber gloves you need for the Half Dome cables. If you don't have these gloves or if you have trouble getting certain pieces of equipment, you might need to adjust the details of your climb accordingly. With all these decisions made, all that's left to do is to head out and enjoy your vacation.

From Planning Vacations to Biodesign

Let's now look at how this vacation planning process worked so that we can see how it's akin to the biodesign process. The first step was identifying the challenge you wanted to address: you need a vacation! Next, you brainstormed a few different possible approaches: the tropical getaway, urban adventure, or backpacking trip. None of these approaches was "right" or "wrong," but you thought about their relative strengths and weaknesses to solve your current problem and decided that the backpacking trip was the best approach.

Then, you got down to planning the details of the trip, using the following abstraction levels to reach your final plan (see also Figure 2-1):

- You began by looking at the big picture—what we call the *system* in biodesign—and decided that you want to take a three-day trip to Yosemite.
- The next level, which roughly correlates to what we call a *device*, was deciding that you want to climb Half Dome.
- The next planning step for your trip was choosing the specific trails that you were going to use, essentially defining the mechanism for your Half Dome "device." These trails correspond to what we call *parts* in biodesign.
- Finally, the equipment list describes the physical pieces you need to buy or make to implement your trip. In our biodesign analogy, these pieces can be considered the *DNA* for the design: they're the actual physical pieces you need to collect and put together to build your system.

Abstraction is an important aspect of biodesign. It allows you to think about and communicate your plans without becoming bogged down in the details all at once. Some useful abstraction levels for biodesign include systems, devices, parts, and DNA. Whether you realized or not, you went through very similar levels of abstraction while planning your vacation.

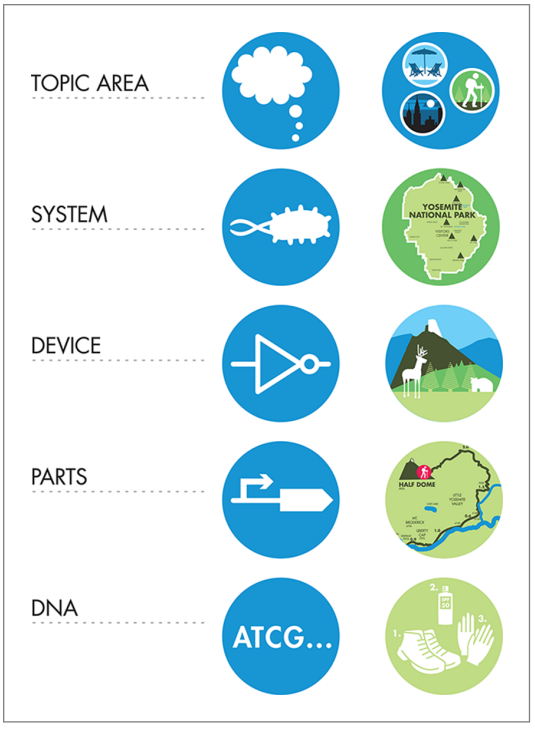


FIGURE 2-1 *Symbols representing different abstraction levels. The thought bubble in the blue circle at the top of the abstraction hierarchy represents the selection of a topic area. The cartoonish cell or virus in the blue circle represents the system level for biological abstractions. The device icon in the blue circle is a Boolean NOT gate, borrowing a device symbol from electrical engineering. The blue parts icon uses the symbols of functional DNA elements, namely a promoter and open reading frame, and the DNA icon in blue shows the four nucleic acids that encode genetic parts. Their parallels in the backpacking analogy from the text are shown on the right.*

There are **some key features to this design process worth spotlighting**. First, nowhere along the way was any answer "right" or "wrong." Each decision was made to best meet your needs. Second, you didn't specify everything at once. For example, you didn't immediately decide how you were going to climb Half Dome; you just provided a general specification that this is what you want to do. This helped to make the planning portion of the trip manageable. Imagine if at the very beginning of this design process you had tried to jump to a parts-level design, deciding on trails to follow for a hiking trip anywhere in the world, as depicted in Figure 2-2. Even if you limited yourself to three-day backpacking trips, you essentially would have been choosing between every hiking trail on the planet, which is obviously impractical. You would have been overwhelmed by the possibilities, and your trip would probably never happen. By using different levels of abstraction, each decision along the design pathway became manageable. Abstractions are powerful ways to work comfortably with less-detailed descriptions during the design process, knowing that you'll be able to specify at the necessary level of detail when needed. Finally, you designed your trip in such a way that it could be modified. If you suddenly found yourself with 10 days rather than 3, you could keep your Half-Dome plan but add another "device-level" plan; for instance, you could add a climb of El Capitan, one of the park's other renowned peaks.

FIGURE 2-2 *Starting in the middle of the abstraction hierarchy. The Yosemite map with the urban background shows the futility of looking for hiking trails before narrowing the search to exclude cities.*

Before we move from planning vacations to planning biotechnologies, there's one more analogous feature to point out. After you decided on your hiking route, you could have created your own map to better visualize your plan or to share it with your fellow backpackers. In biodesign, this map could be considered analogous to a wiring diagram for your genetic system, which is an electrical-engineering formalism synthetic biology has adopted to communicate some design elements of each system. We'll go into the drawing and interpretation of wiring diagrams later in the chapter.

From Design to Implementation

Successful designs will eventually be implemented, and although this chapter focuses on the design portion of synthetic biology, it's worth briefly extending our backpacking trip analogy to show how it applies to the implementation phase, as well.

In planning the backpacking trip, you began with very broad specifications and drilled down until you reached the most granular abstraction level: the equipment list. Note that this is just a list, and not the actual items themselves. The transition from items on a list to the actual tangible objects represents the transition from the final step of the design process to the first step toward implementation. To implement your design (that is, your trip), you'd begin building back up from this granular level until you can head out and enjoy the vacation you've planned. This would involve accumulating the items on the equipment list, perhaps by buying them at the local outdoor gear store or digging through the bins in your basement. Then, when you had all your gear, you'd pack up your bag, head to the trailhead, and start your trip.

For synthetic biology design, the DNA pieces are the "equipment" that you'll need to implement a design. These are the physical snippets of DNA that encode the genes, promoters, terminators, and other genetic components of the designed system. To implement the design, you'll need to obtain these genetic pieces, just as you needed to accumulate the equipment for your hiking trip. There are a number of ways you could get the DNA snippets. These methods include Polymerase Chain Reaction (PCR) and *de novo* synthesis, which are explored in detail in the Fundamentals of DNA Engineering chapter. You might already have some of the DNA parts available in the lab. When you have all the physical DNA pieces you need, you're ready to stitch them together and get to building the system you've designed.

Biodesign Process Overview

There is no single, correct way to design a biotechnology. Depending on what your system specifications are and what you have available already, you could commence in any number of ways. Here we lay out some steps that can help with the biodesign process if you are beginning from scratch. We've found this to be a helpful framework to use and also one that's easy to modify depending on the expertise, timeframe, and goals associated with the project. This design process specifies the details at the different levels of our abstraction hierarchy, from the system-level design to devices, parts, and finally the DNA. Briefly, here are the biodesign steps used:

- Step 1: Identify the challenge area that you'd like to address
- Step 2: Brainstorm approaches to solve the challenge
- Step 3: Decide on an approach
- Step 4: Specify your system at each level of an abstraction hierarchy

Identify an Area and Challenge

You begin by choosing a problem that interests and excites you. Table 2-1 provides a few possible starting points. This table is divided into two categories: Area and Challenge. Area is very broad, referring to an entire issue or discipline, whereas Challenge refers to a specific element of the area that might need to be addressed; the table provides only a few examples of the many possible areas and challenges synthetic biology could address.

Table 2-1. Some topic areas and associated challenges to address through biodesign

Area	Challenge
Food or energy 	People need to eat. Planes, trains, and automobiles need to eat too. One grand challenge for humanity is how to meet everyone's needs in the face of growing populations and increasing reliance on cheap fuel. Current methods of food and fuel production face harsh criticism for their negative impacts on the environment and global food security. How can we use synthetic biology to produce food or energy for the world in a sustainable way?
Environment 	From Love Canal to the Great Pacific Garbage Patch, humans do not have a great track record of taking care of the planet. But finding a way to maintain clean air, water, and land for future generations is key to our survival as a species. How can we use synthetic biology to help clean our planet and preserve its natural biodiversity? Or, if the Earth is a lost cause, how can we use synthetic biology to terraform another planet or asteroid?
Health or medicine 	Microbes have long been known to cause disease, but more recently it has been discovered that they can also underlie good health. How can we engineer microbes or other organisms to aid in the detection and treatment of disease? Are there synthetic biology applications for managing chronic illnesses or surgical intervention or even for maintaining good health?
Manufacturing 	Cells are full of tiny, complex nanoscale machines such as the ribosome, which is an assembly line and factory floor for producing proteins. These proteins have diverse functions, from performing chemical reactions to serving as the structural material for tissues like skin and hair. How can we harness the cell's intrinsic manufacturing skills to produce novel compounds, materials, or structures?
New application area 	So many challenges in the world could benefit from a biological engineer's perspective. Consider issues that you have heard about in the news or those you have confronted in your own life. What entirely new application areas can we imagine for synthetic biology? Think globally or locally, just make sure you think outside the box.
Foundational advance 	Synthetic biology stands on the shoulders of technological advances dating as far back as 35 years ago with the invention of recombinant DNA technology. To ensure the field continues to advance, it needs to develop new, improved tools and technologies to help make the entire process of engineering biology easier. What foundational technology will advance synthetic biology in the years to come?

To narrow things down a bit, it can be useful to pick a single area on which to focus; any area faces multiple challenges, and Table 2-1 provides just a small sampling of the possibilities. When deciding on the challenge you would like to tackle, it is important that you are precise in describing the problem. But at this point, you shouldn't include any specific ideas about *how* you plan to address it. For example, something like "finding a way to eliminate only malaria-infected mosquitoes" will set you on a better path than the more vague "curing malaria," but note that neither description addresses how to implement the idea.

If, at this stage, you find you have a long list of challenges you'd like to address, you can apply some of the questions that follow. These questions can help narrow the

playing field somewhat, allowing you to consider the more impactful and implementable ideas you have while setting aside some of the possibilities that have better alternative solutions or that can't readily be addressed with synthetic biology:

How precisely can I describe the challenge and opportunity on which I'm focusing?
: A clear scoping of the problem you see and the reason you believe you can address it will help with later design decisions.

If my project is fully successful, how big a difference could it make? What concerns will it raise?
: Imagining success will help you to decide if a project is worth doing. An idea that makes a small difference in a huge challenge is valid, as is a project that completely addresses a more minor problem we face. And, you can decide which is more appealing to you. However, a design that would never be adopted for ethical or practical reasons should be considered less appealing. Ethical considerations in biodesign are more extensively considered in the Fundamentals of Bioethics chapter.

What other technologies can I use, or that others have used, to address this area?
: Biotechnology won't always be the right approach to address an issue. If there are other cheaper, reliable, or easy ways to solve the real-world problem you've identified, your idea might be less attractive.

What don't I know about this challenge and my approach to solving it? What either isn't or can't be known?
: Science has informed us a lot about some things (how genetic changes lead to some disease states, for example) and less about other things (how life could survive on Mars). There will be a steep learning curve for almost any project as you begin to dive into the details, so knowing what you could find out and what is still a mystery might help you decide on a project idea.

All these questions are important. Your list of ideas might yield really good answers to some questions but really weak answers for others. Generally speaking, the more precise you can be about answering these questions, the better shape you'll be in for taking the next design steps.

Imagine that you've chosen to address the challenge presented by arsenic-contaminated water, which is a problem in Bangladesh and West Bengal. Because there's no easy way to determine whether the water is contaminated, people living there have been drinking arsenic-contaminated water and getting sick. In fact, a student team tackled this challenge in a synthetic biology design competition called International Genetically Engineered Machines (iGEM). Next, we walk through that team's project to provide a concrete illustration of biodesign in action.

The iGEM team wanted to design an easy way to establish whether water is contaminated with arsenic. It decided to engineer a cell to create some sort of easily detectable

signal when there is arsenic present. This plain-language system description, "a cell that can create a human-detectable signal when arsenic is present," is a good example of a system-level description appropriate at this point in the design process. It is precise but not overly mechanistic. As the project idea becomes more refined, it will be further specified, but here it's helpful to stay at this high level of abstraction.

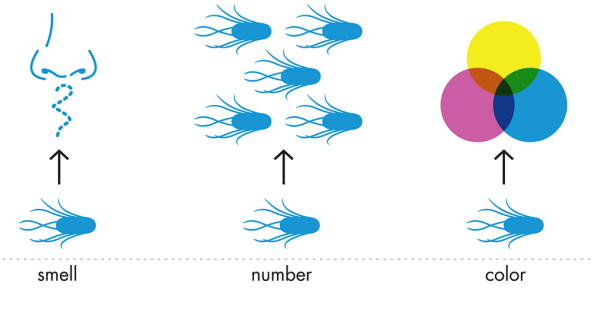
You can also see how applying the earlier list of questions would have helped the iGEM team confirm that it chose an appropriate topic. The problem is clear: otherwise undetectable arsenic in drinking water poses serious health repercussions. The team also precisely defined the approach, which is relatively straightforward: build a system that generates a human-detectable signal in the presence of arsenic. If it successfully builds an easy-to-use arsenic detection system, it could significantly improve the health of those in Bangladesh and West Bengal, especially given that other technological solutions to this challenge are expensive, thus limiting the broad distribution of those solutions in potentially contaminated regions.

Brainstorm Solutions to the Challenge

After you've identified the issue you would like to address, it's time to begin thinking about different ways that you can solve it. What's tricky at this stage is that you don't yet want to become bogged down with the details; we're still talking about big picture ideas that could provide a framework for the more detailed design that will come later. There are many possible solutions to any problem. Ultimately some might end up being better than others, and you will explore these strengths and weaknesses a little later as you decide on an approach to pursue. At this point, though, you can really let your imagination run wild.

For our arsenic-detecting system, a few possible solutions might be:

- A cell that produces an odor when exposed to arsenic
- A cell that begins growing and dividing very quickly when exposed to arsenic
- A cell that displays a distinct color when exposed to arsenic
- *Your idea here!*



If you find that you are having trouble coming up with ideas at this point, perhaps you're too focused on the details of how you will implement them, which might be blocking your creative juices. Notice how none of these ideas indicate how the cells will detect

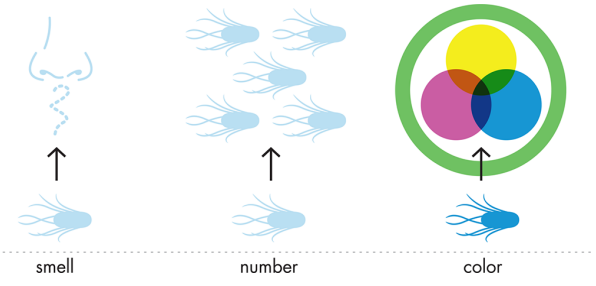
arsenic, or even what kind of cell we're talking about. Remember that we're thinking BIG PICTURE here, not details!

Decide on an Approach

Now it's time to pick one (or maybe two) of the ideas you listed in the previous step and begin specifying a design. At this point we are just narrowing down the options to eventually find a single approach that meets the challenge you've defined, but remember that we are not locked into any approach yet. Design is an iterative process. As you think about your design and as you build some prototypes, you will almost certainly run into hurdles. In some cases, you can overcome these hurdles, but in other cases it might be most effective to return to an earlier phase of the design process and reconsider your approach entirely. This is not a failure; it's an optimistic feature of how design and engineering work. There is a good solution out there and you can keep looking for it, and, as you do, you will be learning something from the process. As you weigh ideas, you can return to the questions from "Identify an Area and Challenge" on page 24 to guide your thinking. Questions such as, "how precisely can you define the challenge?" and "how big a difference can your project make?" will clarify the intent and specifications of your work.

When you've decided on a project idea, like the arsenic-detecting project idea from the iGEM team, it's time to make a few choices. The iGEM team had particular design specifications to meet. They wanted to ensure that whatever signal their cell produced was easily detectable by whoever was doing the testing, and they also wanted to ensure that the response was relatively quick, so the person collecting water didn't have to wait a long time to know whether it's okay to use the water from that source. To choose between an arsenic detector that turned the water a particular color, that produced an odor, or that grew at different rates, the team could consider the following:

- The expected feasibility of implementing their system based on knowledge of existing biological parts and approaches
- The flexibility of the approach because a system that is generalizable could be used for other applications later on
- The expected success of the design, where simpler designs usually have a greater chance of success



In weighing the three approaches, some things become clear. Scent-production in the presence of arsenic might not be a detectable enough indicator, and there could be a lot of other competing environmental smells. A change in growth rate would probably be too slow to detect and could also be caused by a number of other factors other than arsenic. So, of the three project ideas, an arsenic-induced color change seems the most attractive.

Notably, at this point the nitty-gritty details of the design are still not specified. It still hasn't been decided what color to produce or how arsenic will be sensed; instead, only the general direction has been chosen. Some of the specifics of implementing the approach are in place, but we still haven't become bogged down in the details—that comes next!

Specify Your System with Devices, Parts, and DNA

With the high-level system design in place, it's time to start specifying how it will work, delving into the details of implementation. However, rather than jump right to the genetics of it, we'll go step-by-step through different layers of abstraction.

In the previous step, you crafted a broad description to outline how the system should act in response to different stimuli and environmental conditions. To build such a system, you will need to get down to the DNA level and decide on a *chassis*, which is the term used to describe the organism that runs your genetic program. To help with the system specification, there are two intervening layers of abstraction: devices and parts, as illustrated in Figure 2-3.

Comprising one or more parts, a *device* converts some "input" that we can specify into an "output" we can detect or that can feed into another device. For the arsenic-detecting example, for instance, the iGEM team knew it needed some way to detect the arsenic and some way to generate color in response. The team could have implemented its system with one device, namely one that senses arsenic and creates color. Alternatively (and, in fact, what they chose to do), the system could be implemented with two devices: one to sense arsenic and another for color generation. The input for the arsenic-sensing device is arsenic and its output is a signal that the color-generating device can detect. The output for the color-generating device is a color. These very broad specifications are sufficient to define the devices needed to implement the arsenic-detecting system and we leave the details of how each of these devi-

ces work individually and how they communicate with each other to consider in the next abstraction level.

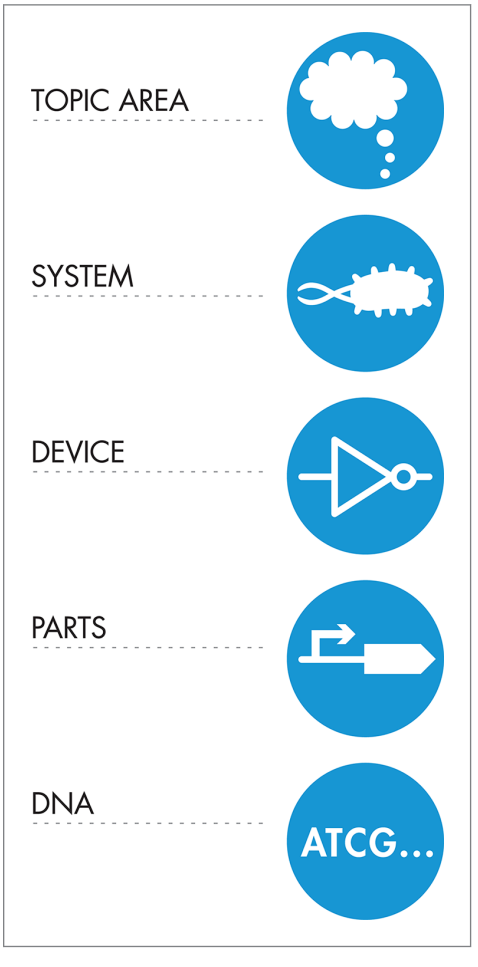


FIGURE 2-3 *Extended abstraction hierarchy for biodesign. Abstraction can support complex system design. This abstraction hierarchy includes "topic area" as the top level, emphasizing the need to select a broad challenge to focus on. The abstraction also includes the entire system, which would include a plain-language description of your design idea. The system then can be broken down into specific devices that are made up of certain parts that are built from DNA.*

In some cases, devices might carry out a logic function similar to those commonly used in computer science and electrical engineering. We describe Boolean logic functions in more detail a bit later, but briefly, examples include an AND gate, which only generates output if *all* the required cellular signals are present, and an OR gate, which generates output if *any* of the desired signals are present. Because you can implement these input, output, and logic devices in many ways, they are higher in the abstraction hierarchy than the parts that make them up, which we'll explore next. In fact, it's possible (even desirable) to come up with a list of devices for your system idea without thinking about how you can make them.

Parts are sequences of DNA that encode basic biological functions, such as promoters, open reading frames (ORFs), and ribosome binding sites, which are described in greater detail in the Fundamentals of DNA Engineering chapter. To achieve a device's required function, synthetic biologists assemble the device from simple parts. Sometimes, a DNA sequence can be a part and sometimes it can be a device, depending on how it's operating in the system. Generally, though, devices can be defined by the biological consequences they have in the system, and they are most often made up of multiple parts. The arsenic-sensing device might require a protein part that binds to arsenic, as well as a promoter part to start transcription of that protein coding sequence, whereas the color-generating device might include a different protein part that produces some colorful chemical signal. That device could use the same or a different promoter part as used in the arsenic-sensing device to express the color-generating protein that is the output of the device.

We consider devices and parts as different levels of abstraction because this is a helpful way for us to design and build systems. However, the borders between devices and parts are not like the borders between the elements on the periodic table. It's possible to define an atom by its number of protons, but there is no universal physical law to say what is a "device" and what is a "part." Indeed, you might come across cases in which you can consider the same biological component either a device or a part. As we have emphasized multiple times in this design context, there is no "right" or "wrong" answer. Rather, these different layers of abstraction can be very useful for a designer as she is considering complex systems, and that utility to the designer is the most important aspect of this abstraction hierarchy. So, don't allow yourself to become bogged down with whether you're talking about a device or a part; the point of these different terms is to help *you* with your design. The next few examples will illustrate the utility of these different labels.

System-level design

We'll focus on two components of system-level design: system behavior and chassis selection.

The early steps you followed in the biodesign framework helped you to decide on a system's behavior. For the arsenic project idea, the iGEM team defined the overall desired behavior as follows: arsenic causes a visible color change. This is a "plain language" description of the system. A second way to communicate a system's behavior is to use a *truth table*, a basic logic tool widely used in electrical engineering and computer science. A truth table conveys how a system should behave in the presence or absence of inputs. For the arsenic-detecting system, the truth table is very simple: if the input (arsenic) is absent, there should be no output (the color shouldn't be produced); if the input is present, the output should be on (color should be produced):

Input	Output
Absent	Off
Present	On

This logic can also be expressed a few other ways; for example, where false corresponds to "off" and true corresponds to "on":

Input	Output
False	False
True	True

Or, you can use math symbols, where "–" corresponds to "off" and "+" corresponds to "on":

Input	Output
-	-
+	+

Or, you can use binary code, where "0" corresponds to "off" and "1" corresponds to "on":

Input	Output
0	0
1	1

Remember, these all represent the same basic logic for the system, just using different vocabulary. As you build more complex systems, you might need more complex truth tables. Truth tables can also be useful when designing complex devices. We provide a few examples in the description of logic devices that follows.

A second aspect of system-level design is determining which host organism, or *chassis* (see Figure 2-4), you will use to run your genetic program. You can use many different organisms and cell types as chassis for designed systems, including bacteria, yeast, viruses, and mammalian cells. Two of the most commonly used chassis for synthetic biology are the bacterium *Escherichia coli* and the yeast *Saccharomyces cerevisiae*. These two hosts are well characterized, safe, and behave relatively reliably, making them suitable chassis for many designed systems. But depending on the application, it could be wise to choose a host with specific growth characteristics or that has a naturally occurring protein that would be useful to the design in mind. Going back to our backpacking analogy from the beginning of the chapter, Yosemite was the chosen system but if you didn't like steep climbs or crowds, it wouldn't be the

right system choice, and you might want to rethink your decision to hike Yosemite versus the Appalachian Trail, or to choose the high country in Yosemite over the valley.

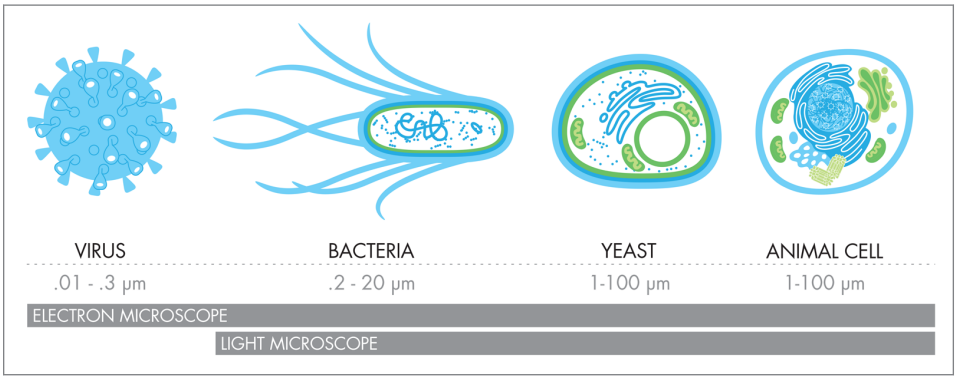


FIGURE 2-4 *Select a chassis. Synthetic genetic programs are typically housed in relatively well-understood laboratory systems and cells, including those shown here. Chassis are arranged from smallest (left) to largest (right), with approximate sizes and tools for visualizing them shown below the icons.*

Device-level design

After you have specified your system-level design, you can begin working on the device-level implementation—climbing Half Dome from our backpacking analogy. Again, remember that you don't need to specify the exact physical mechanisms by which your devices will work. Instead, as shown in Figure 2-5, we'll use general labels such as "arsenic detector" and "color generator" to highlight *what* they do, but not necessarily *how* they do it. In other words, at this point you can name any type of device you want, and you don't need to worry about how it works or even if it really exists yet. It might seem as if you are pulling devices out of thin air, but natural biological systems are so diverse that we can likely create any type of device we want based on existing biological parts or small modifications thereof.

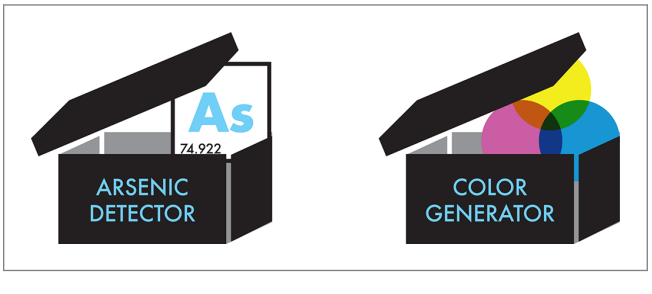


FIGURE 2-5 *"Black box" depictions of two devices. The arsenic-detecting and color-generating devices are shown in a way that illustrates what they are meant to do rather than how they work.*

Another consideration, besides the type of devices needed to build the system, is how many devices would be best. For example, the iGEM team making the arsenic-detecting system could specify its system with two devices: an arsenic detector, which takes arsenic as an input and provides a signal as the output, and a color-generating device, which takes as input the signal from the arsenic detector and produces color as the output. The team could have achieved the same system function with a single device, one that takes arsenic as an input and produces color as an output, but such a device would be highly specialized and might only be useful for their one system. By separating the system into two devices, the team defined more generalizable devices that might be useful for other applications in the future. It is helpful to think about defining devices that will have long-term utility in a variety of applications.

When a system requires multiple devices, it can be useful to use another tool from electrical engineering to sort out the details of the design: a *wiring diagram*. Electrical engineers graphically represent circuits using wiring diagrams, which explain how parts and devices connect to one another. They can be literal, with the connections representing actual wires and the parts representing specific electrical components such as transistors and capacitors, or they can be more abstracted and conceptual illustrations of how a design should work. For our biological implementation, the wiring diagrams are mostly conceptual, because we do not have actual wires to connect the different components of our system. Making such a wiring diagram can be a useful approach to work out designs and communicate with other synthetic biologists. These wiring diagrams are like the map you might draw for your backpacking trip to climb Half Dome; you can both refine your plan and share the details with others.

For our arsenic-detecting system, the wiring diagram will be pretty simple, as shown in Figure 2-6.

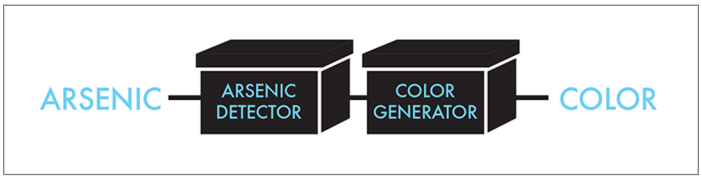


FIGURE 2-6 *Wiring diagram for an arsenic-detecting system. The arsenic-detector device, still a black box, is connected to the color-generator device, also a black box, illustrating that a color output is made in the presence of an arsenic input.*

As Figure 2-6 makes clear, this is simply a visual representation of what we've already described verbally; arsenic is the input to the arsenic-sensing device, which produces

a signal as the output when arsenic is present. That signal then acts as the input for the color-generating device, which produces color as the output. These diagrams can become significantly more complex as the systems become more complicated. You can see some additional examples and variations in the description of logic devices that follows.

Logic devices

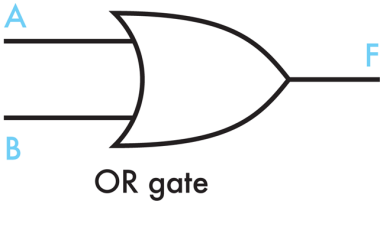
Truth tables are a way to express how a system responds to different combinations of inputs, which we call the Boolean "logic" of the responses.

Imagine two compounds, A and B. Depending on the identities of A and B and the goals for a particular system, we might design different responses to the combinations of these two compounds. For illustration, let's further elaborate the arsenic-detecting system we've been considering and imagine that we'd like to sense a number of compounds in the environment. To keep things general, we'll call the compounds A, B, C, and so on. Maybe A and B are each toxic on their own, so we would want our system to respond if either of them is present. Alternatively, maybe neither is toxic alone, but if they are both present, they react to create a toxic compound, C. In this case, we would only want our system to respond if both are present.

We can represent these two cases in a truth table. **Truth tables represent all possible combinations of inputs and specify a single output for each combination**. The input columns don't change between the two tables. Instead, the outputs vary depending on the logic of the system and which input is present. As an example, here is the truth table for the system that should be triggered in the presence of *either* A or B:

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

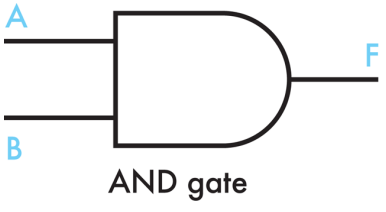
Such a truth table represents the operation of an *OR gate*, which in wiring diagrams is given the particular symbol shown here:



Here is the truth table for the second system that should be triggered only if *both* A and B are present:

Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

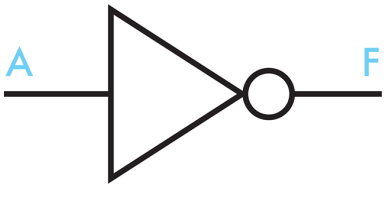
This next truth table represents an *AND gate*, which is represented by a subtly but meaningfully different symbol from that of the OR gate:



AND gate

There are still other kinds of logical combinations of inputs and outputs that are different from those described by the AND and OR gates. For instance, a Boolean *NOT gate* is one which inverts an input signal into its opposite. It is shown in the following truth table, and symbolically in the figure that follows (note that the NOT gate takes only one input):

Input A	Output F
0	1
1	0



NOT gate

With this visual language and some simple combinations of the logic gates, it's possible to build quite sophisticated systems, both electronic and living. For example, the following truth table for a *NOR gate* is simply the opposite of the table for the OR gate:

Input A	Input B	Output F
0	0	1
0	1	0
1	0	0
1	1	0

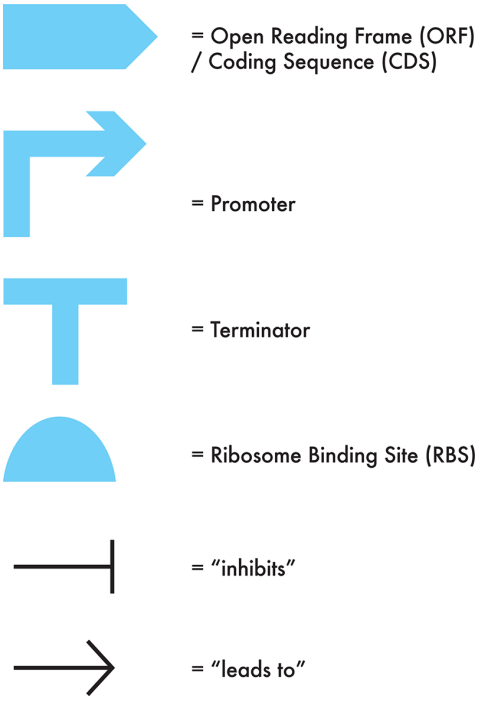
NOR gate

Visually, the NOR gate is represented as an OR gate with a circle at the right tip, adding a NOT to its function and making it a NOR gate.

One of the key research publications that launched the field of synthetic biology built a NOR gate to operate in a living system. The work was from the laboratory of Dr. James Collins at Boston University. His research group built two NOR logic gates using genetic parts and cross-wired them to create a *latch*.

Parts-level design

When you're designing at the level of parts, you begin (at last!) to discuss *how* your devices can actually carry out their assigned task. Let's return for a minute to the planning of our hike. In that case, there were certain trails and pre-existing ways to go about realizing the basic plan. Similarly, when planning a genetic program, there is a toolbox of biological "parts" to build devices. These parts include ORFs that encode transcription factors or fluorescent proteins, sequences that enable RNA polymerase to start transcription ("promoter parts"), and a variety of sequences that encode gene expression regulators, to name just a few. The number of parts that exist in the world is nearly endless and completely overwhelming. There are, however, catalogs of biological parts, such as the Registry of Standard Biological Parts , to facilitate this *parts-based approach* to biodesign. The parts and devices you can find listed in these open catalogs have been documented with the intent that you could reuse the parts for new biodesign projects.



To illustrate parts-level design, let's return one last time to the arsenic-detecting system. This system will need something that detects arsenic and some way of conveying the presence or absence of arsenic to the next device in the system, namely the color-generating device. There are a number of ways to build an arsenic-detecting device. One straightforward way would be based on an arsenic-sensitive protein found in nature. **Implementing an arsenic-detector device could be accomplished through a combination of three simple parts**, namely a promoter (upward right arrow [↱]) to start transcription of the device's DNA, a ribosome binding site (top half circle [◠]) to start translation of its mRNA, and an ORF (long play button [▭]) to encode the arsenic-sensitive protein. Similarly, the color-generating device could be built from a simple promoter part, a ribosome binding site part, and an ORF that leads to a color change in the solution or the cell itself.

These devices, as designed, will sense arsenic or generate a color, but in order for the information from one device to be transmitted to the other, we need to be somewhat clever. **To enable the arsenic detector to "talk" to the color generator, the output of the arsenic detector must influence the input for the color generator**. One common approach is for the output of the first device to be a protein that can regulate the rate of transcription of the second device. In this case, the arsenic-binding protein is also a transcriptional repressor that, in the absence of arsenic, binds to the promoter for the color-producing gene, blocking its expression. Arsenic, when present, binds to the arsenic-binding protein, causing it to change conformation so that it can no longer bind to the promoter, which frees the promoter for transcription of the color-producing gene. You will see many variations on this theme as we continue to explore synthetic biology systems.

Verbal descriptions like this can be hard to follow, so we have another visual language to express how a genetic circuit works, as demonstrated in Figure 2-7.

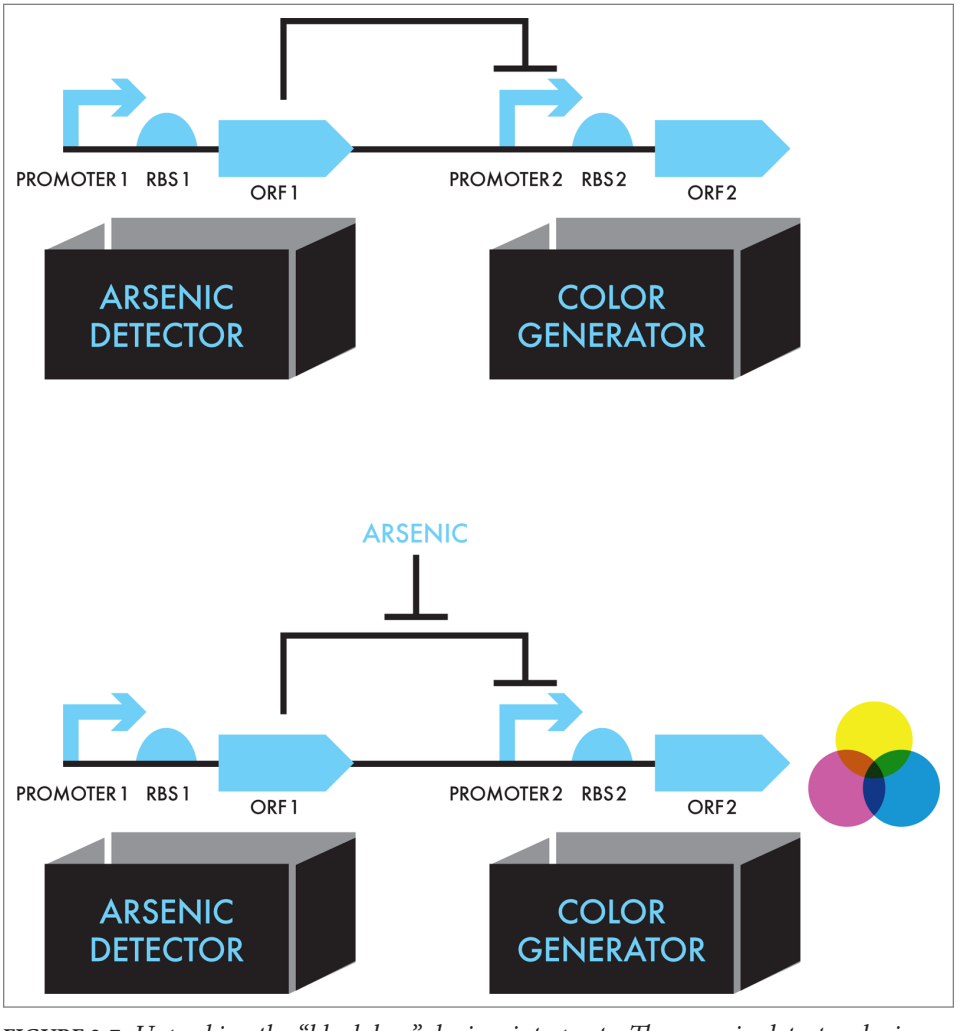


FIGURE 2-7 *Unpacking the "black box" devices into parts. The arsenic-detector device and the color-generator device are each composed of a promoter, an RBS, and an ORF. In the absence of arsenic (top panel), the product of ORF 1 inhibits promoter 2, and therefore the color generator is "off." Relief of inhibition in the presence of arsenic (bottom panel) frees the color generator device to turn "on."*

This diagram in Figure 2-7 shows that in the absence of arsenic, the output from the arsenic-detecting device will inhibit the function of the color-generating device. When arsenic is present (as it might be if the well water is contaminated), the inhibition is relieved, in this case through the action of another inhibitor, allowing the second device to turn on color production.

As with the design choices made at every other level in the abstraction hierarchy, here, too, there are decisions to make based on the particulars of the system you're trying to design. For example, all promoter parts are short DNA sequences where transcription factors and RNA polymerase bind, and they'll always be placed upstream of a gene, to transcribe the DNA into an mRNA strand that a ribosome will then translate into a protein. However, there are many different promoters, some strong, some weak, and some that can be regulated in interesting ways, so picking the appropriate promoter is also an important aspect of the design process. The arsenic-detecting system required that the arsenic-detector protein be present constantly, so the promoter for that ORF was selected because it could remain "on." This type of promoter is referred to as *constitutive*, and there are a number of constitutive promoters in the Registry. There are also a number of promoters in the Registry that would be inappropriate for this system. For example, some promoters, called *inducible promoters*, are active under specific conditions—the presence of a sugar, for example, or the number of cells growing nearby. You would not want to use an inducible promoter for the arsenic-detecting device because this would lead to expression of the arsenic-detecting protein only when certain compounds or conditions are present. Such inducible promoters might be just right, though, to build other systems.

After the final desired sequence is specified, there are several strategies for building it. The details for these assembly strategies are a subject for the Fundamentals of DNA Engineering chapter, but briefly, most DNA assemblies involve a combination of *de novo* DNA synthesis and molecular biology tools such as PCR, restriction enzymes, and molecular cloning.

What's Next?

In this chapter we have introduced the fundamentals of biodesign and illustrated the different layers of abstraction and symbolic representations that can be useful as you work toward your own designs. One idea we've emphasized is that there are no "right" or "wrong" answers as you go through the design process. Going back to our analogy from the beginning of the chapter, design is a journey. **You will certainly experience some false starts and wrong turns as you work to develop your biological designs, but these are learning experiences, not mistakes**, that will help you better understand

your system and inform your design process as you go through the cycle of design-build-test iterations.

Of course, design is just the first step and we'll end this chapter with a short exploration of the next stage, namely the implementation of your design. Returning again to our backpacking analogy, implementation of this trip would involve buying or making the necessary equipment, packing your bags, getting yourself to the trailhead, and starting your hike. During the hike, you might spend some time just looking at the ground in front of you, making sure you don't trip on a root or perhaps noticing a single wildflower; other times you might take a break to look up and appreciate the entirety of your surroundings, or to take in a sweeping vista when you get to the top of Half Dome. You might go off-track once or twice (or more), or you might realize that the original route you planned wasn't actually the one you want to take and change your plan. You might become too dehydrated and not even make it to the top of Half Dome. Although such a failure might be disheartening, you will have learned something about your design that will help you improve your plan for your next trip. Regardless of the outcome, you'll have a whole new wealth of knowledge that you could apply to your next iteration of the design.

Buying or making the hiking equipment corresponds to buying or making the DNA sequences required for our biodesign, and you could think of packing your bags as inserting these DNA sequences where they're needed in order to construct the designed organism. Then, getting to the trailhead and starting the hike is the testing process of our design. We can conduct different types of tests to look at individual elements of the design, like the small wildflower, and we might also conduct tests to investigate the system's overall behavior, like looking at the view from the top of Half Dome. It's also important to remember that it is completely normal and expected to take a few wrong turns, change our minds about the route we want to take, or sometime to realize that we won't reach our desired outcome. In all of these cases, **we can use what we have learned along the way to improve the next iteration of our design**.

Just like our hypothetical backpacking trip, you can consider biodesign and its associated implementation as a journey. During that journey we end up looking at our system from a variety of angles and levels of abstraction, we find ourselves using

different kinds of tools to realize our goals, and we undoubtedly take some wrong turns before getting on track. As long as we pay close attention to how our system is behaving, we'll be learning along the way, which will improve both our understanding and our next design iteration.

Additional Reading and Resources

- Danchin, A. Scaling up synthetic biology: Do not forget the chassis. *FEBS Letters* 2012;586(15): 2129-37.
- Galdzicki, M. et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology* 2014;32:545-50.
- Gardner, T.S., Cantor, C.R., Collins J.J. Construction of a genetic toggle switch in Escherichia coli. *Nature* 2000;403(6767):339-42.
- Marchisio, M.A., Stelling, J. Automatic Design of Digital Synthetic Gene Circuits. *PLOS* 2011; DOI: 10.1371/journal.pcbi.1001083.
- Oremland, R.S., Stolz, JF. The ecology of arsenic. *Science* 2003;300(5621):939-44.
- Website: Arsenic Detector iGEM project (*http://bit.ly/arsenic_detector*).
- Website: Registry of Biological Parts (*http://parts.igem.org/Main_Page*).