Universal Central Processing Unit and Evolving Transforming Computers

(Universal CPU) Computers that physically change its hardware and software during runtime.

Mitchell Kwok
2008,2009

Universal Central Processing Unit (universal CPU): a CPU that can physically change its circuit gates during runtime.

Evolving Transforming Computers: computers that physically change its hardware and software during runtime.

The universal CPU is a computer chip that is adaptable and changeable and has the ability to physically change its circuit gates during runtime. Every single electronic device or machine that you buy in the department store uses a different computer chip. A smartphone uses a different chip from a desktop computer and a pocket calculator uses a different chip from a video game console. Besides selling more copies, chip manufacturers do this because they want to build a customized CPU that processes computer instructions, quickly, for each electronic device. For example, a video game console has a special CPU, called a GPU, to process video and 3-d model rendering, while a desktop computer has a different type of CPU to run common software programs like spreadsheets or word processors. If you try to apply a desktop computer CPU to run a video game console, the video game frames will display slowly on the TV. In some cases the video game program won't even run. This is why chip manufacturers make it standard procedure to build optimal CPUs for each type of electronic device.

Different CPUs for different electronic devices

A Universal CPU is one chip that can be used on all electronic devices or machines, regardless of their size or shape. What's unique about this universal CPU is its amazing ability to physically change its circuit gates and electronic circuitry during runtime. This novel approach will enable the Universal CPU to create an optimal customized chip for any given electronic device.

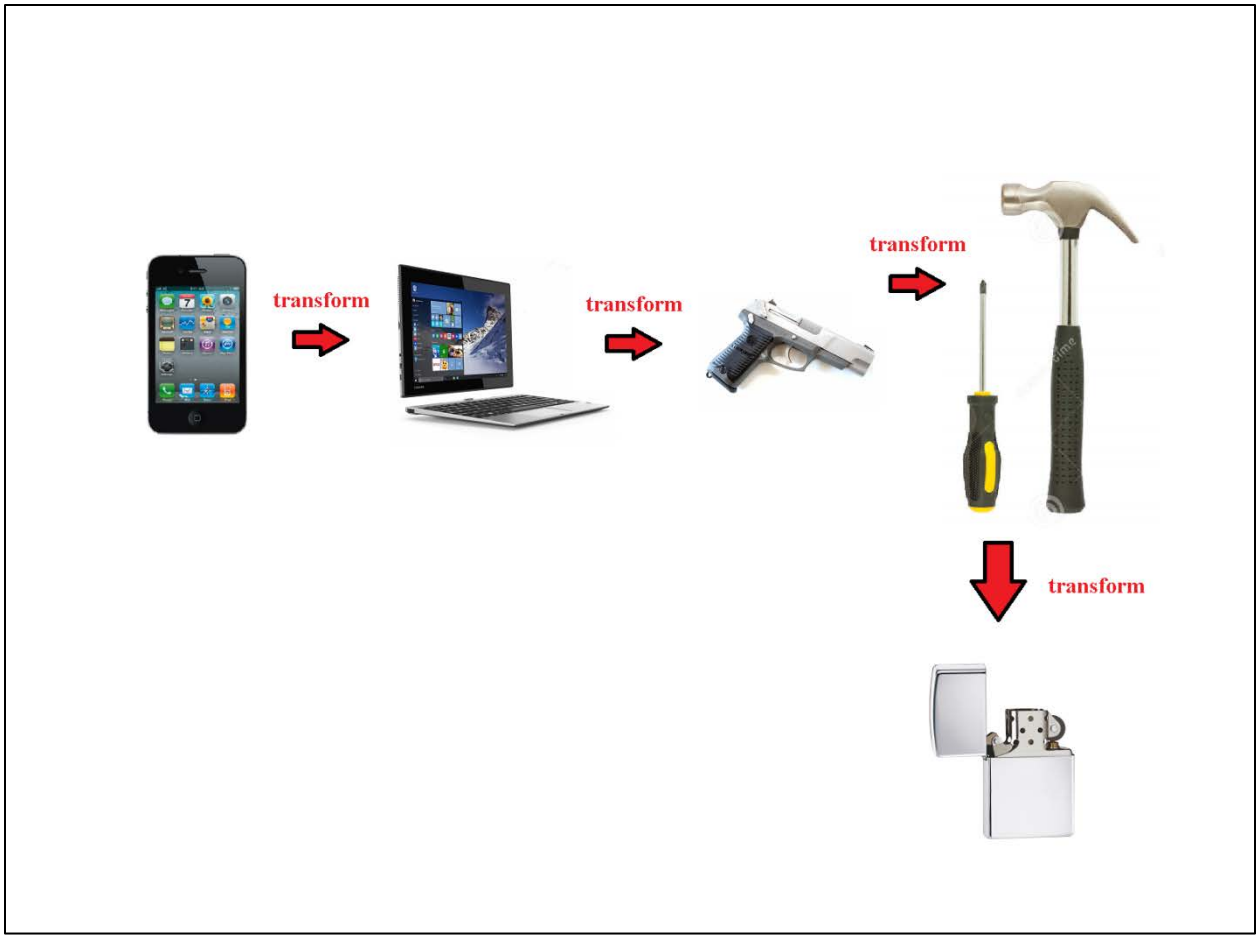
The Evolving Transforming Computer (ETC) is a computer that can physically change its hardware and software during runtime. Actually, the computer can physically change three aspects: its hardware, its software, and its CPU.

Let's take a closer look at how this is done. The initial machine being analyzed is a car. Let's imagine the user wanted to change the car into a desktop computer. At a molecular level, the machine will transform its physical molecules and re-position them to form a desktop computer. The trade secrets to this method will be revealed later on in this book.

Not only does the A.I. have to change the machines' hardware, but also, its software. The cars' software morphs into the Windows Operating System. The circuit chips in the car also morphs and changes into an Intel CPU, commonly found in most desktop computers.

Since a car is physically larger than a desktop computer, the excess molecules comprising the car are put into an atom reserve area.

The Evolving Transforming Computer can change into any object; it doesn't have to be an electronic device or machine. For example, it can transform into non-intelligent objects or common tools, like a hammer, a gun, a lighter, or a bed.



This technology is also an evolving computer that evolves its hardware and software in an intelligent way according to the user's needs and wants. Most people are just tired of Apple coming out with new iphone editions each year. The consumers are given the daughting task of purchasing a successor iphone each year. This is very expensive and inconvenient. What if you had a smartphone that can evolve and make itself better year after year?

Let's say you bought an iphone4 and it has evolving and transforming capabilities. As a result, annually, your smartphone will transform its hardware and software and self-upgrade its features and capabilities. In year 1 of owning the iphone, it turns into iphone5. The year after that it will turn into iphone6. By the time you own the iphone for 6 years, the initial iphone4 will transform into iphone10.

It works exactly the same way with any other electronic device. If you apply this technology to a car, the car will evolve year after year based on the user's preferences. Here is a desktop computer. It evolves its capabilities and features as time passes. The Universal CPU will evolve its circuit gates. It will add, delete or modify circuit gates during runtime. Other transformation will happen, such as increased ram capacity or minimized circuit gates in specific regions of the CPU. If the Artificial Intelligence finds a better design for the CPU it will replace the old design with something better.

Let's take a closer look at the hardware. The hardware will also go through significant physical transformation. Let's say the desktop computer was bought as a 1980's computer. Every year, new hardware design will be used and external hardware is added to the system. For example, a mouse will be added to the system because it's a useful tool to communicate with the computer; or a printer will be added to the system to allow users to print important documents and letters.

As the desktop computer evolves over time it will go into "uncharted territory". New devices like holographic projectors will be added to the hardware. Maybe a non-physical keyboard with pressure capabilities is added to the system to replace the bulky keyboard. Maybe.. the computer might evolve into a wallet sized object that you can carry around in your pocket.

Let's look at the software. Imagine the desktop computer was bought as a 1980's computer. This means the operating system will be from the 80's as its initial starting point. The operating system will evolve its capabilities by turning from one windows operating system into its successor. In less than a decade the windows1 operating system will evolve into the windows 10 operating system.

If we let this Evolving Transforming Computer evolve the desktop computer, the end result will be the most optimal desktop computer that a technology company can design and build, in terms of hardware, software, and CPU. This future computer is the fastest and most dependable computer ever developed. Not to mention the capabilities on this desktop computer is superior compared to anything on the market.

Let's say you wanted to produce the smartest and fastest robot on the planet. We can use the Evolving Transforming Computer on a humanoid robot. The A.I.'s main objective is to make the robot smarter year over year. This robot will evolve its physical body and internal software each year and it will get smarter and smarter as time passes. This "thing" will eventually evolve into the smartest robot in the Universe. Another important fact to point out here is that this robot evolved in the fastest time possible and every generation design was executed optimally.

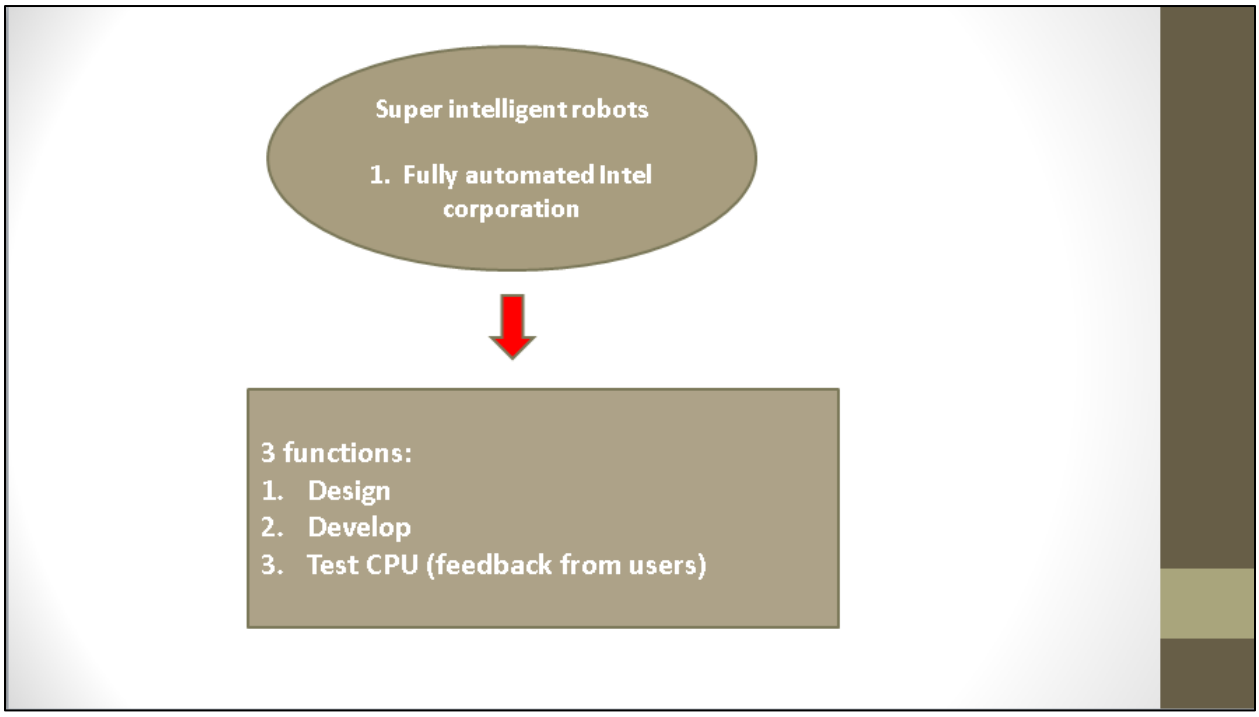
--

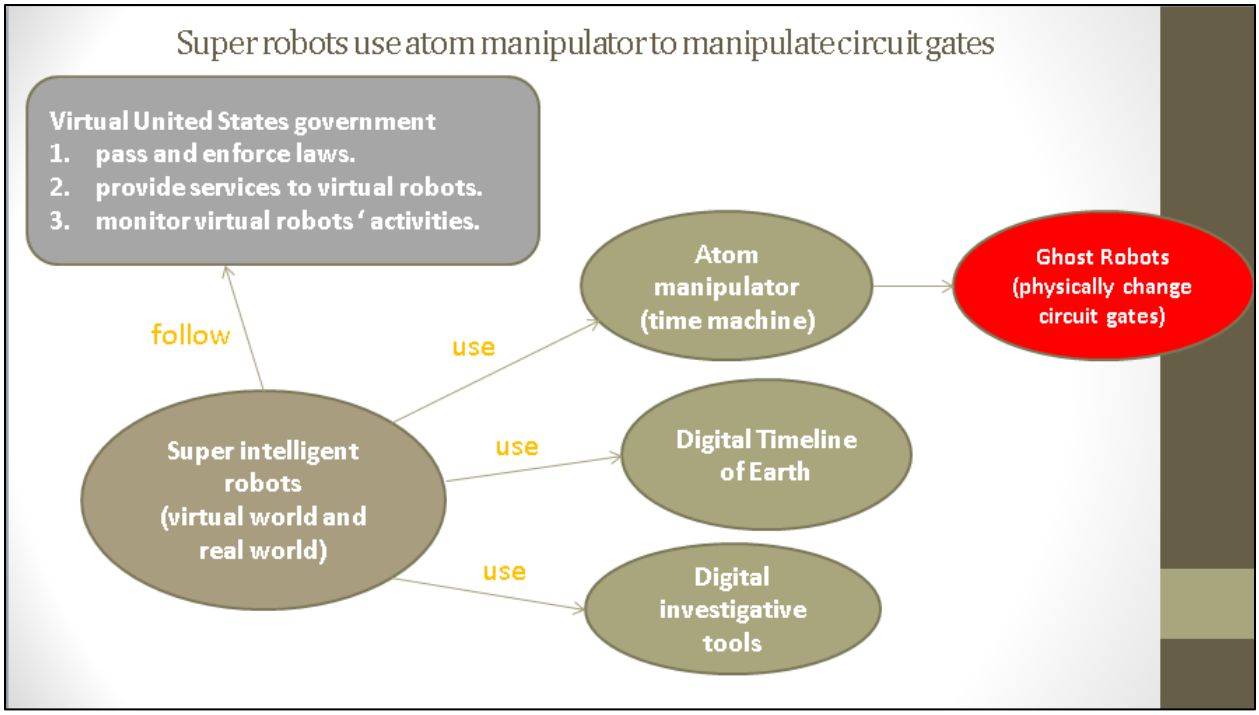
How does the technology work?

The last video I made on Youtube was on the Ghost Robots. This book is actually part two of the ghost robots. I recommend the viewer to watch the Ghost Robot video before proceeding forward.

The most complex task for Ghost Robots is the ability to manipulate individual atoms and molecules. A computer CPU is a very good example because it comprises tiny circuit gates. Having the ability to manipulate these circuit gates shows the ghost robots' ability to manipulate tiny atoms.

The A.I. is made up of super intelligent robots. The robot makes thousands of copies of itself and structure these virtual robots into a company like setting. This company, which only exists inside the virtual world, is similar to a chip company like Intel or AMD. They specialize in designing, manufacturing, and testing computer chips.





The Super intelligent robots control the atom manipulator to generate "intelligent pressure" and intelligent/controlled force fields. This in turn, creates tiny ghost robots, which are non-physical robots, to manipulate the CPU. These ghost robots are really tiny, about the size of several atoms or molecules.

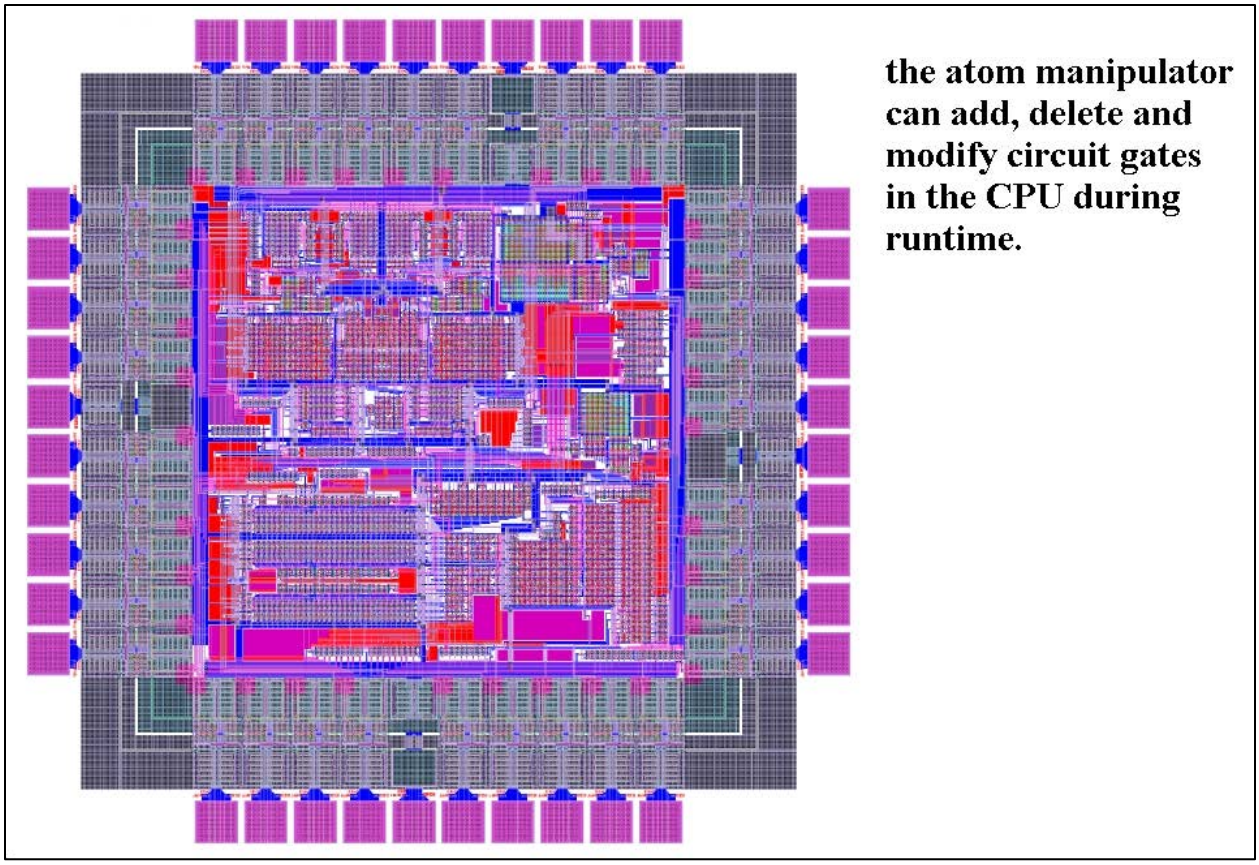
Here is a regular CPU found in most desktop computers. The A.I. has to physically manipulate these circuit gates during runtime. This includes things like adding circuit gates, deleting circuit gates, or modifying circuit gates in various parts of the CPU.

For example, we can upgrade this CPU by doing a few things:

1. The ghost robots can add more RAM. This will give the chip more memory capabilities to store processed data.

2. They can add more integrated circuits, depending on the type of CPU. This will allow the computer instructions in the task queue to be processed quicker.

3. They can modify the circuit gates in the CPU. Referring to FIG. 45A, here is an example. This circuit gate isn't optimal. In order to optimize the circuit gates, the A.I. has to minimize the circuit gates. The ghost robots can do this by rearranging the pathways. Referring to FIG. 45B, here is the circuit gates minimized. This modified circuit gate performs the same functions as before, but there are less circuit gates present. This is a method chip companies use to make their CPUs cheaper and faster.



Each generation of the computer CPU is called an iteration. For simplicity purposes an iteration, from start to finish, is set to 1 year. The computer CPU is upgraded after each iteration. In less than 6 years the Intel4 chip went through 6 transformations and turned into the Intel10 chip.

Let's take a look at each iteration. The primary function of the super intelligent robots is 3 things:

1. designing the computer CPU.

2. implementing the computer CPU.

3. and testing the computer CPU (and considering feedbacks from customers).

Everything that Intel does, these super intelligent robots also have to do. This includes designing, implementing and testing their chips.

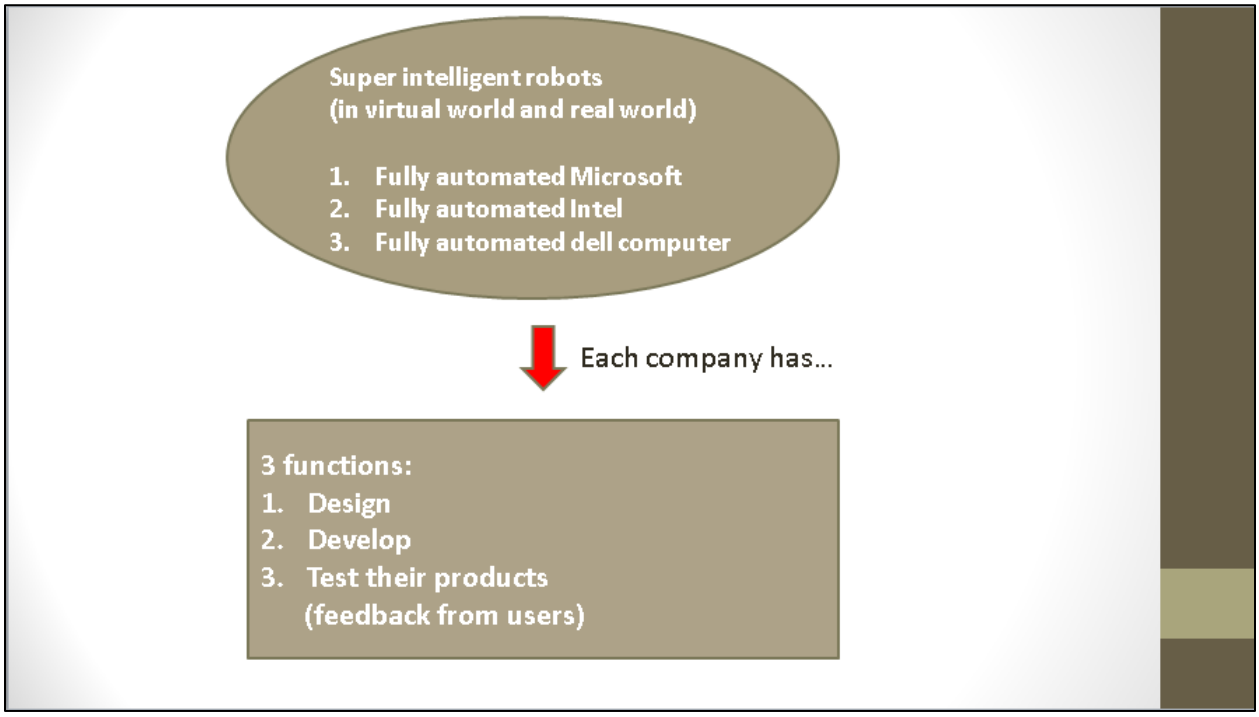
These super intelligent robots serve as "a fully automated Intel corporation".

--

Let's take a look at the evolving computer. The super intelligent robots are responsible for emulating 3 companies: Intel, Microsoft, and Dell computer. Intel is in charge of designing and developing the CPU,

while Microsoft is in charge of designing and implementing the computers software, which is primarily the operating system, and Dell computer is in charge of designing and developing the computer's hardware.

These 3 companies have to work together, in both the real world and the virtual world, to design and develop the best computer.



Referring to FIG. 42, The super intelligent robots are using the atom manipulator and the digital timeline of Earth to physically change the atoms of the computers' hardware and software. You can read my book on the atom manipulator to understand how the process works.

Notice the physical changes of the computer as it evolves from one generation to the next. The computer starts of as one big bulky machine. The next year, it separated the computers' hardware into parts like a monitor, keyboard, mouse, and speakers. In the second year, it changes into a folding tablet and evolved into something smaller and more powerful. Finally, after 3 years of evolution the computer transforms into a tiny cube that doesn't have a physical monitor or keyboard. It uses holographic projections and pressure technology to replace bulky hardware like a monitor or keyboard.

The super intelligent robots are constantly taking the computer and modifying its software and hardware according to their consumer's usage behavior. If the computer is used for playing video games or watching movies, the design of the CPU will be modified to make the computer run faster. If a consumer is complaining about the bulkiness of the computer, the super intelligent robots will make the computer lighter and smaller. If the operating system has a security problem, the A.I. will provide a

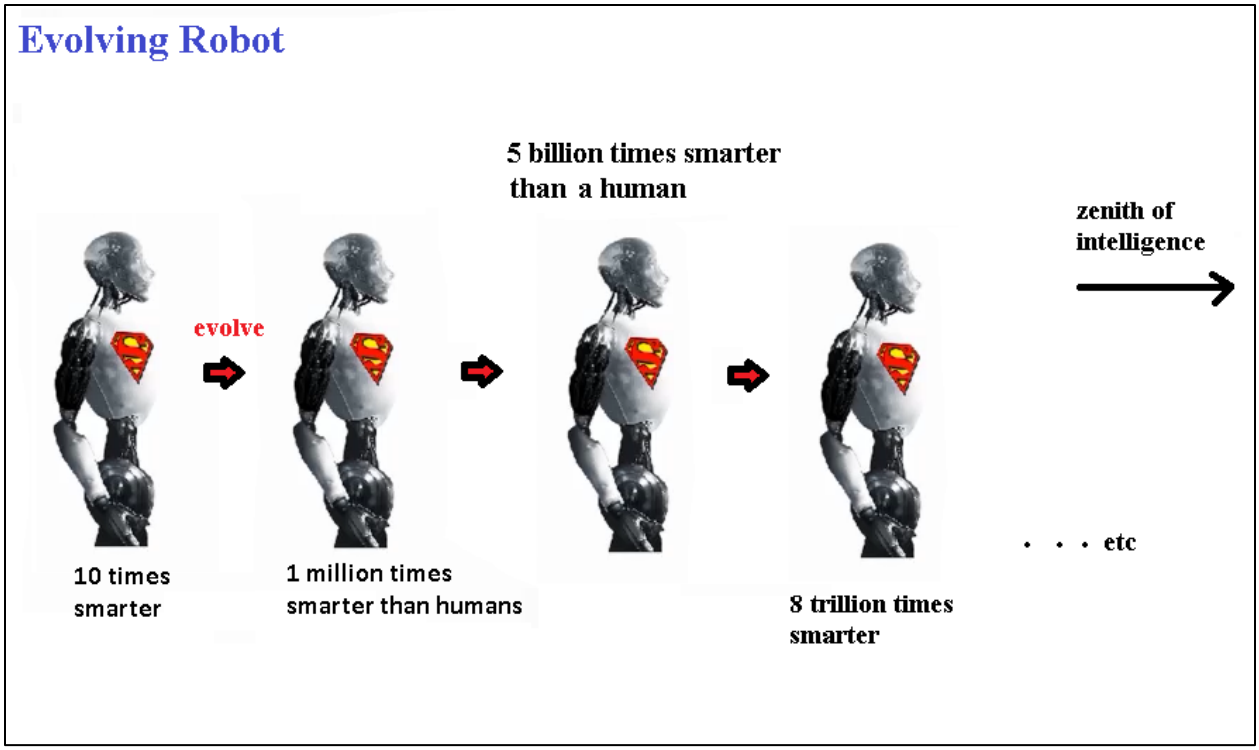
software patch to fix the problem. If Intel has a better design for their CPUs, they will replace the old design.

This is how an evolving computer should work. Some A.I. scientists think that they can build a self-evolving software program or genetic A.I. program that can evolve on its own. This method is pure science fiction and can never be implemented. Their approach is impractical because there's no intelligence guiding the evolutionary process and this method would require billions of years of evolution to produce anything meaningful.

The key is the super intelligent robots. Their intelligence is what allows this computer to evolve and transform its software and hardware.

Let's take a look at another example. Here is an evolving robot that evolves its intelligence. Let's say you wanted to build a robot that can evolve its intelligence as time passes. The ultimate goal is to build the smartest robot in the universe. If we use modern genetic programming or self-evolving software, it won't work. This process will take billions of years to produce anything significant.

According to my method, we set up a robotic company inside the virtual world. The A.I. in the virtual world emulates 3 companies: Intel, Microsoft, and Dell computer. Their responsibilities are to modify the robot, in terms of software and hardware, to make it smarter year after year.



If we run several generations on the robot, it will eventually evolve its intelligence exponentially. As you can see, after 4 generations, the robot is 8 trillion times smarter than a human being. We want this robot to evolve and transform until it reaches the zenith of intelligence. Another important fact to point out here is that this robot evolved in the fastest time possible and every generation design was executed optimally.

The most important component of this technology is the super intelligent robots. Without that component, without their intelligence, there is no way the robot can evolve its intelligence on its own.

The super intelligent robots have to be smart enough to replace all 3 companies and their employees. Often times, I use the terminology, fully automated Microsoft or fully automated Intel to describe this technology.

Speed of each generation or iteration

The speed of each iteration depends on the work speed of the virtual robots. Inside a computer time is meaningless. We can define time inside a virtual world. We can say 30 years inside a virtual world is equivalent to 1 second in the real world.

In this book, the iteration for each generation is set at 1 year. We can set the iteration to 2 months or 5 hours or 1 second.

Let's imagine that 1 iteration is set to 1 second. In less than 10 seconds, that's 10 iterations, the windows1 operating system can evolve into a windows10 operating system.

The problem I have is how do we force these super intelligent robots to work inside a virtual world for 30 years to write an operating system. We know that each robot is self-aware and is capable of performing tasks at a human-level. But how do we force self-aware robots to do work?

This problem will be addressed in my next topic, which is called: the nature of work in 2100. I will attempt to predict how people live and do business in year 2100. I will also provide possible solutions to the problem.

As stated before, in order for intelligent beings to have a wide variety of skills and knowledge it has to be self-aware and must graduate from college. Using machine learning or deep learning or modern A.I. techniques will never produce a fully automated Microsoft or fully automated Intel corporation.

--

Conclusion:

This is my last invention I'm currently working on. The picture on the left is a modern CPU found in common desktop computers. The picture on the right is my CPU. Notice something odd about my CPU, there's nothing there.. there's no circuit gates.

This CPU is non-physical in nature and uses atoms and light to generate Boolean algebra. This technology is also called a Universal CPU. I proposed 3 types of universal CPUs: 1. a physical CPU that can morph its circuit gates during runtime (as described in this book). 2. a semi-physical CPU that has external ram and temp registers. 3. a non-physical CPU that uses light and atoms to generate Boolean algebra.

For the past 10 years I was trying to find an alternative way to make a CPU. In college I took courses in integrated circuits and machine translation. So, I have some background knowledge about how CPUs work. One day, I had a dream about boarding a UFO and was given the privilege to take a peek at how alien technology works. I remember taking the computer apart and looking at the central processing unit, the CPU. What I saw shocked me. I saw an absence of any physical integrated circuit or gears or wires. There was nothing there except a transparent glass. Inside the glass I saw light zipping back and forth like a highway system. That's when I realized that the glass was somehow processing computer instructions using light interactions.

This dream inspired me to find alternative ways to build CPU's that is different from conventional methods. I have been trying to design a universal CPU that processes computer instructions using either light interactions or atoms in air. The RAM part is solved by using electron spins, the up spin is 0 and the down spin is 1. Either that or the RAM can be an external hardware.

A modern CPU

My universal CPU (non-physical CPU or semi-physical)

Remember, this CPU uses air to process Boolean algebra and not physical circuit gates. The more computer instructions, aka Boolean algebra, are needed to process, the more air space is needed. This method is fast, cheap, reliable, and environmentally friendly.

For the light interactions, a specific math equation is used for each elemental circuit gate. The half-adder, full adder, the not-gate, and the or-gate, are all represented by specific math equations. I can combine these elemental circuit gates, by combining equations together, to construct any computer chip or Turing machine during runtime.

The Boolean algebra is not based on monitoring the light or atom interactions, but is based on the outcome of the math equation (constructed during runtime). Since the math equation represents future outcomes of light and atom interactions, it will ultimately determine the exact "path interactions and outcomes". Only the input and outcome of the equation is observed. No equation is processed in the computer. Based on this, the A.I. will guess which and how many circuit gates were used and the temp register digits to ultimately re-construct the real output. For example, the input might be adding two numbers (which are represented by light and/or atoms) and the output is the answer. This CPU is scalable. The input can be a complex math equation and the answer might be 8 doors.

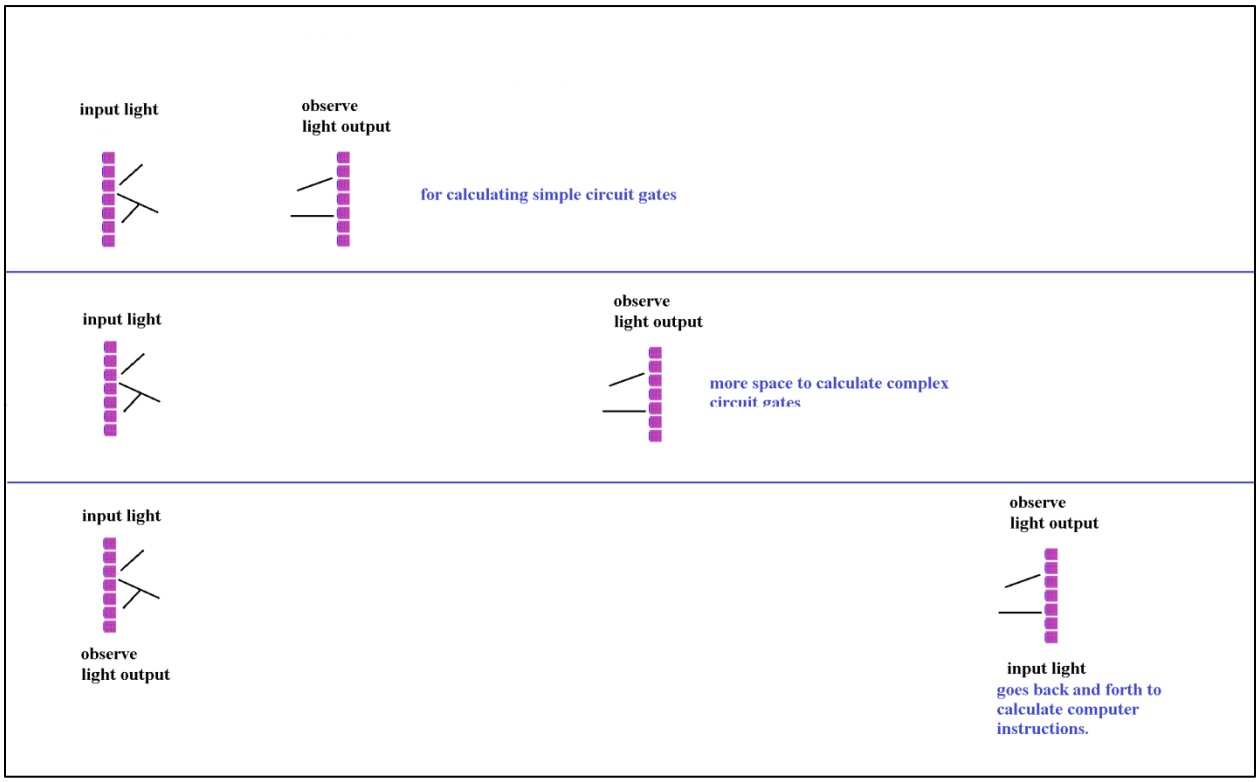
The process goes like this:

1. provide the light and atom interactions (this is the input). One example is add 2 numbers.

2. observe the outcome.

3. based on the outcome, reconstruct the answer. This is done by guessing the digits in the temp caches and guessing the elemental circuits gates used.

I think of light interaction as ball interactions on a pool table. If a ball goes into a hole at a specific time period the variable is a 1. If the ball doesn't go into a hole at a specific time period, then it's a 0.



Maybe a form of machine learning, like deep learning, can be used to optimize the CPU.

Wither this new CPU can run faster than conventional CPUs remains to be seen. The important thing is that it's a different type of CPU with different abilities and properties.

I still have a few more problems to solve before I can complete this invention. A patent or book will probably be filed with the U.S.P.T.O when I'm done.

This invention is only a design concept, almost like a rough sketch, and there is no guarantee it will work.

The nature of work (part 2)

Another idea is to build an artificial universe. This method is kind of like the matrix movie. Inside this artificial universe are intelligent beings that evolve into entities that have human level intelligence. We can insert our agents into these intelligent planets and force said intelligent beings to do work for us.

This method is despicable and unethical because we are forcing unsuspecting intelligent beings inside a computer to do our work. On the same token, it's a fair trade off. We give them life, and in return, they give us what we want. We want intelligent beings that have a wide range of skills and knowledge to do our work.

As stated before, in order for intelligent beings to have a wide variety of skills and knowledge it has to be self-aware and must graduate from college. Using machine learning or deep learning or modern A.I. techniques will never produce a fully automated Microsoft or fully automated Intel corporation, even if A.I. scientists had a million years to accomplish this task.

Condition robots to work in harsh environments

Another idea is to condition the robots to do unwanted work. We can send it off to elementary school, so that it acquires basic common sense skills and knowledge. After that, we can teach it to work in a restaurant repeatedly. All of its pleasure senses will be attached to the majority of what it learns. If the robot is a chef, then the learning will primarily include cooking and cleaning chores in a restaurant. This type of conditioning will force the robot to like what it was designed to do.

I remember a shocking article I read from a newspaper a few years ago about a criminal who committed a botched bank robbery and was sentenced to 20 years in prison. When he got out, he immediately committed another crime and was sentenced to another 15 years in prison. When a reporter asked him why he committed the second crime, this is what he had to say, "I've been in prison for so long that living in prison is my life. I feel more comfortable in prison than in the real world".

This story is just one example of how people can be conditioned and brainwashed to do certain bad things. Working in a factory for long hours may seem cruel, but if a human works in a sewing factory for 20 years, they are conditioned to like their job. If they were to do some other work, like cook in a restaurant, they will feel uncomfortable and pain.

We can use this method to build robots that like their occupation and gladly work in factories for long hours at end in return for some benefits.

The programmers can also rewire the robot's senses to like bad things and to dislike good things. If the robot is working in a junk yard, we can program the robot to like the smell of sewage or garbage. Humans on the other hand, have offensive smells innately graphed into us by our creators; and garbage smells bad.

We can program the robots to like working. If humans were forced to eat good food like ice cream and lobster roll all day, they would gladly do it for free. If humans were forced to go on vacations or do the things they like to do, they would gladly do them for free. This method might yield unintended consequences because the programmers are building robots that are totally different from human thinking. But this is a good method to force them to voluntarily do work.

Using this method we can also re-condition the robots to avoid violence. The majority of disruptive behaviors and crimes are committed by human males. This might be because our creators made man and woman differently, in that man is created more likely to commit violent crimes. The programmers can program feminine behavior into the robots so they are innately prone to avoid violence. Feminine behaviors can also be learned by society and through teachers in the public school system.

Another idea is to tell the citizens, robots or humans, they have to do work to pay off taxes owed to the federal government. The funds will be given to the government to pay for basic services like running the government or paying for police stations. Computers that the robots are using to run virtual reality worlds and the internet is also paid forth by government funds.

As a token of appreciation and a sense of patriotism, the robots might voluntarily work and contribute to pay for these services.