JUL 9, 2017

My Career at IBM

JACK LEE

Jack is a volunteer at the CCNY Archives. Before retiring, he worked at IBM for over 28 years. His articles have over 100,000 views.

Introduction

My 28 years career at IBM is divided into two parts. The first 9 years were spent in the IBM CPD division at the Fishkill and Kingston facilities. The second 19 years were spent at the IBM Research facility in Yorktown Hts. NY. My intent in writing this part of my life is to help the next generation understand the past and what it was like working in a Corp. environment. I also don't want this to be all about me but about some technical subjects that may be of interest to the public. Please take the poll at the end and let me know how I did. Alternatively, you can send me comments at the end of the hub; there is no need to sign up to HubPages to comment. (Just enter your name as a guest). I also added a list of terminology at the end.

-Jan. 2016

Background

My career at IBM was a long and productive period that coincided with the transition from mainframes to PC's. My story is not unlike many other individuals that worked at IBM. I was recruited right out of college and moved around as the needs of the company required. At its peek, IBM employed over 400,000 people worldwide. I considered myself very lucky to be part of this great corporation. The many colleagues that worked together to produce extraordinary products is truly amazing. It is a living testament to the fact that the collaboration of the many can be greater than any individual. Our talents were varied and our combined efforts along with a Corporate culture that supported our work has lead to real progress for society and profits for the shareholders. This unique culture was set in motion by Thomas Watson who founded the company and continued under his son Thomas Watson Jr. The three basic beliefs include:

- Respect for the individual
- Service to the customer
- Excellence must be a way of life

I will tell my story from my perspective. I've divided my career into two chapter to reflect my two completely different careers. The first being a design engineer in the Communications Product Division and the second as a researcher in the prestigious IBM Research Division. I must specify at the outset that my story may be a limited view of history since I was not privy to all the information and backgrounds of the various projects. I do offer an unique perspective on the various projects from a first hand account. However, I don't want to make this only about me. I want to tell the story of an IBMer, one who is among thousands that made up the Corporation.

My First Share Of IBM Stock

My Beginning

In January of 1974, I graduated CCNY with a BS degree in Electrical Engineering. Along with six other fellow graduates, we were chosen to start our careers at the IBM facility in Fishkill, NY. I was hired to join a new group of development engineers to work on a FS system. (Future System). This was to follow the recently launched IBM System 370 series. My assignment was to design a Loop adapter for the new system. It was an ambitious project that required a large team of engineers of which I was proud to play a part. It was a learning experience for me and an eye opening experience coming from a school environment. I had to learn all the Corporate procedures and computer systems for development.

One memorable event as a new hire was attending the orientation meeting. A senior VP came to the session and gave us a pep talk. It really made us feel important and being part of a successful Corporation with a bright future. His main message was instilling the three basic beliefs. I remember being really impressed by his humble attitude and sincerity. He was a cheerleader for the IBM way and it was infectious.

As a regular employee, there were many benefits one of which was a stock purchase plan that allow us to set aside a small portion of our salary to buy IBM stocks at a discount. It took me approx. 6 months to accumulate enough to purchase my first share. I retained that certificate to this day as a memento. I recall it was worth approx. $600 per share in 1974.

My First Success

Coming right out of school and joining a large organization can be intimidating. I was unsure of what to expect and what was expected. I dove into the new project and learned as much as I could. I found out quickly that the real world is no way close to what we studied in school. What was important is knowing how to approach a new problem and finding a viable solution.

My first task as an Associate Engineer is to design the logic to implement a CRC checker for a loop adapter. This cyclic redundancy checker is a simple but effective method to insure data transmission over any communication line. To verify the design, I needed to learn a new language called APL. It is a very powerful language that is efficient at computing matrix calculations. Without going into technical details, it was truly a learning experience and one that I tackled head on. My program was a success and it gave me the satisfaction and the confidence to move on to other bigger and better projects. Unfortunately, the FS project was canceled after just one year after I joined IBM due to the ambitious nature of that project. However, it was not a total loss for me or for IBM. It gave me the training to learn on the job and prepared me for the next one. My training included learning to use the design tools such as BDL/S and BDL/C to create the circuit designs and working with ALD and computer outputs to verify those designs.

UC-1 Project

The CPD division was one of several divisions of IBM. It's main task was to produce communication products for business machines. Our next project was to develop a micro controller named the UC-1. It's predecessor the UC .5 was the workhorse for a slew of products including printers and terminals and communication products. It has a new Architecture with a unique set of instructions (or microcode). The microcontroller is the "brain" of the system that performs the calculations.

Our group was a small team of approx. 10 engineers and one manager. There was a team technical leader who interfaces with other groups within IBM including the Architecture group and the Manufacturing facilities both located in other IBM facilities. In addition, we worked with a local support group who helped with the tools and IT infrastructure needed to produce the product. Another small team of programmers that were responsible for writing the test programs that will eventually verify our design completely. That is, they create test codes that will exercise the processor in all operating conditions. This was no small task since our final product was only as good as the test programs that verified it.

My small part in all this was to design the ALU. It is the arithmetic logic unit of the CPU. It performs the basic operation of the CPU. Having learned the tools of design, it became a tedious design and simulate and update cycle which involves very detail and precise work. It is also my first introduction to the LSSD design unique to all IBM products. The LSSD is a strict logic design rule to make any design easy for testing. Once the design is complete, an automated program takes over and creates a RIT that will be sent along with the design to the manufacturing house. This RIT is a set of codes that will exercise the final product through test signals applied to the I/O ports. The codes will determine if the product was manufactured correctly as designed. This LSSD was necessary because of the VLSI technologies that puts more and more dense circuits onto a silicon chip.

The project was estimated to take three years from start to finish. We completed the design on schedule and the final product was a huge success. It worked beautifully and integrated with a line of communication products that used the same architecture.

A Detail Of The Chip I Help Design

Tusken Project (IBM 8100)

IBM to internals stands for I've Been Moved. In 1978, our group was relocated from Fishkill to the Kingston facility. Our follow-on project to the UC-1 was the next generation of VLSI micro controller nicknamed "TUSKEN". It was to be an ambitious custom bi-polar design comprising of four chips with embedded PLAs. Each chip would be the equivalent of 5000 transistor circuits. This new controller will be the processor behind the new IBM 8100 System. A series of new computers with a totally new Operating system, not one but two, DPPX and DPCX.

In the same mold of previous projects, I was part of a new small team comprising of 10 engineers reporting to a new first line manager. I was promoted to Staff engineer and assigned two new hires to work with me. I was put in charge of the design for one of the four chips.

It was also at this time that I embraced the concept of system software simulation as part of the design process. The challenge of designing a VLSI chip was the long manufacturing process. It takes approximately 6 months to complete a design cycle. If there were any mistakes or fixes, it would take another 6 months to implement. This makes it imperative to get the design right the very first time. Our strategy which has been used successfully in the past was to create a hardware model of the design. This would allow us to "debug" the design in a system environment to verify the design before committing the final design for manufacturing. This process was time consuming and required skills such as soldering and rewiring boards and using a scope. In fact, we divided the team into 2 shifts where we could debug the system more rapidly.

As a parallel effort, I pushed for using our software simulator to integrate the 4 chips. Typical logic design involves small test codes to verify each piece of the design. In computer lingo, this is called a "cut". However, to connect multiple cuts into a system simulation is rare and very hard to do. It involves over night runs on the MVS computer and involvement of tool designers to chase down system bugs. I was one of the first in IBM to create a system simulation of a whole processor comprising 4 cuts. This effort lead to the discovery of some early bugs and shorten the total design cycle. In the end, I co-wrote a paper that was published in the IBM System Journal of Research and Development titled "A VLSI design verification strategy" in July 1982. I presented it at an internal conference held at the IBM Research Center at Yorktown Hts. Unknown to me at the time, I would return there a year later to work on some new exciting projects.

It was at this time that Tracy Kidder wrote a book called the "Soul of a new Machine" which detailed the design of a 32bit processor at Data General. The stories were not unlike what went on at IBM behind the scenes though not quite as dramatic. It is worth a read.

An INTEL designer came up with a rule of thumb regarding IC density. It is known as Moore's law. Basically, the density of transistors will double every two years.

Moore's Law

The Final Product

The Success We Achieved

The above photo shows the comparison of the final VLSI module along with the test boards that were the equivalent design using standard IC circuits. Notice the reduced size and density. Four of these modules made up the "brain" of the IBM 8100 System.

Lessons Learned

Looking back now years later, I couldn't help but be amazed by what we accomplished. The small teams with assorted talents were able to achieve great results. I would never imagine I could do the things I did alone. It took a team effort and comradery to work. The whole is really greater than the sum of its parts. Some other lessons learned were the following.

- Details do matter. Paying attention to small details makes a great difference when dealing with VLSI design.

- Learning on the job is more important than classes and books. Many of the tools and utilities and smarts were learned just by daily interaction with colleagues. If we needed some expertise, we seek them out and usually found the answers. As an example, when we started debugging the Tusken processor, we created a hardware model of the design. We had a computer rack and we used a digital oscilloscope to examine the signals. How do you find a bug? We were never taught this in school or lab classes. A senior engineer on our team taught us how. The trick is to put the test program into a tight loop by modifying the instruction set. Once the program is running, we can then use the scope and the clock to determine each step the computer is running and zero in on the problem. This simple technique is what we did to find 99% of the bugs.

- Good management is crucial to success. It is not enough to have great engineers and designers. A good team must include a capable manager who can support the team and manage the resources effectively.

IBM PC (Side Note)

In 1982, IBM created the IBM Personal Computer for the masses. IBM offered a special discount to all it's employees. I was one of the first to place the order for my own use. I truly believe that the PC was going to revolutionize the world. Never in my wildest dream did I know that it would take off as it did. The three major programs at the time were MS Word (word processor), Visicalc (spreadsheet) and Adventure (game). My original system had 64K RAM, 2 floppy disk drives, a dot matrix printer and the software. Total cost with 15% discount amounted to $5000.

The IBM PC was a game changer. It puts the power of computing into the masses. We are all the beneficiary. I will have more to say on this topic in the next chapter.

Summary

My early career in IBM was unusual in the sense that I was lucky enough to be on two winning teams. The statistics are that most projects do not make it to the finish line. By design, we were told off the record, IBM creates multiply projects that work independently and in parallel. Only the best of breed survives. My success average of 66% (2 out of 3 projects) is pretty amazing.

In the case of the 8100 system, even though we brought it to market successfully, it did not do well in the real world. The system was burdened with two competing OS that were incompatible and the documentation were too burdensome. In the end, the emergence of the IBM PC sealed the fate of the 8100 system.

Some Related Information

- A VLSI Design Verification Strategy.
  A VLSI Design Verification Strategy. on ResearchGate, the professional network for scientists.

- IBM - Archives - Valuable resources on IBM's history - United States
  IBM Archives: Valuable resources on IBM's history

- List of IBM products - Wikipedia, the free encyclopedia

- The Human Brain - A Personal Thesis
  A personal journey in writing my memoir exposes some insight on the human brain.

A Simple Poll On My Topic

Glossary Of IBM Terminology

In any large organization, there is a lingo culture and IBM is no exception. In technology, we tend to use many acronyms to help communicate. The following is a short list of terms that you may find helpful when reading my hubs.

- ALU - Arithematic Logic Unit. The part of the processor that does all the computation.
- APL - a programming language used by scientists for the interpretive nature and for precise mathematical calculations especially with matrix manipulations.
- ALD - a large computer printout that is a diagram view of logic circuit designs.
- Assembly code - The most basic instructions to a computer.
- Bit - A basic unit containing the value of "1" or "0".
- Bug - a problem detected in computer hardware or software. The term "debug" is used to find and correct a bug.
- Bomb - a description of a computer run that failed due to some huge unexpected error.
- BDL/S - The computer language that describe a logic design in detail.
- BDL/C - The control language that is used to program a test case to exercise the logic design.
- Boot - a system operation that initialize the computer from power up.
- BUS - a parallel set of signals that is part of a computer interface.
- Compiler - A program operation that takes a high level programming code and creates the corresponding assembly codes to be run by the host computer.
- CPU - Central Processing Unit. Part of the computer that is the "brain".
- CRC - Cyclic Redundancy Check. An algorithm that insures proper transmission of a series of data across communication channels.
- Dump - A large printout that contains the memory values of a program. It is used by programmers to help find a program bug.
- DASD - disk storage.
- Core Memory - the large computer memory bank.
- DOS - Disk Operating System - the first IBM PC operating system created by Bill Gates and Microsoft.
- Floppy Disk - A computer storage unit for the PC.

- FORTRAN - A computer language for main frames.
- FRU - Field Replaceable Unit. A computer component that can be serviced by a Field engineer.
- IBMer - An employee of IBM.
- IC - Integrated circuits.
- I/O - Input/Output - refer to the computer interfaces between the processor and the rest of the system. Input being keyboards etc. and outputs being printers...
- JCL - Job Control Language. The basic instructions to run a main frame computer program.
- Job - A computer program that is submitted for processing on a main frame computer.
- Latch - a basic computer design component that remembers a state "1" or "0".
- LSSD - Level Sensitive Scan Design. A design system that is architected for easy testing after the manufacturing process.
- MVS - IBM main frame operating system.
- PC - Personal computer.
- PDS - A powerful computer programming language developed for internal use by IBM Research members.
- PLA - Programmable Logic Array. A part of a computer circuit that is very efficient in implementing logic and can be optimized to reduce physical chip size.
- PLI - A programming language for business systems.
- RAS - Reliability, Availability, Serviceability.
- Register - A set of latches in a computer processor that helps execute the instructions.
- RIT - A release tape that includes all the design components and instructions to make an IC chip.
- ROM - Read Only Memory. A part of the computer system that is hard coded.
- TSO - Terminal Shared Option. A computer console that allows multiple users.

- VLSI - Very Large Scale Integration. A computer chip that is highly dense and integrates many components into one silicon chip.
- VM - a Virtual Machine system.

Navigation (My Life Story)

Table Of Contents

Previous Chapter

Next Chapter