Computer Architecture and Amdahl's Law

Gene M. Amdahl

My educational background has never included any training in the field of computing, so all of my design activities have been based on my experience and the necessity of solving current problems. Consequently, my computer architecture contributions will largely be autobiographical.

I was raised on a farm in eastern South Dakota and attended a one-room grade school for all eight grades. We didn't have electricity until I was a freshman in high school, so my technical experience was limited primarily to mechanical equipment. I enrolled in South Dakota State College (SDSC) in the fall of 1941 in mechanical engineering, but decided it was not the field for me. So I took a potpourri of courses in math, chemistry, electrical engineering and physics. World War II intervened early in my freshman year, and I joined the Navy as an Electronic Technician, teaching electronics. Returning to SDSC in 1947, I selected physics as my major, graduating in 1948.

I received a Wisconsin Alumni Research Foundation assistantship and began that summer in the field of Theoretical Physics at the University of Wisconsin in Madison. This was an unusual time in physics, for they had just discovered "strange particles" in late 1949, and the name "meson" had not yet been proposed. At that time two other graduate students and I were assigned to determine if a force between nuclear particles proposed by a Japanese physicist could adequately describe the simplest 3-body nucleus, Tritium (Hydrogen 3). We worked for 30 days using an 8-digit desk calculator, and a slide rule to hold two more most-significant digits. We mapped the energy of the system for all relevant ranges of the parameters, but we couldn't quite achieve a stable state. We had found the proposed force to be inadequate, but I had found the means of calculating to be even more inadequate! I then began to think about how the computing could be done better. The University had no information on computers in its library, no courses in computing and no computers, save for an electronic analog computer in the Electrical Engineering Department.

This is a picture of me in front of the entrance hall of the house I built so I could finish my undergraduate degree. I even had to make the front door! The date of the photograph is 1948.

My major professor, Dr. Robert Sachs, recognized my dilemma and arranged for me to get a 2-month summer job in 1950 at the Aberdeen Proving Grounds. My assignment there was to program "super-sonic flow about a 3-dimensional body". The instruction set was that of the EDVAC, then under development. I wasn't given any introduction to programming, or to the structure of the computer. I did not complete the programming during the 2 month period; I also heard that the development of the EDVAC was dropped because the mercury delay line was unstable due to temperature build-up when operating. I was not enamored of the EDVAC structure because the use of fixed point with a limited word length required a lot of rescaling to maintain reasonable precision. As I returned to Wisconsin I formulated a 3-address floating point structure, trying to make it as simple as possible, and to use technologies that were commercially available. I chose a magnetic drum for main storage, with recirculating registers to minimize the use of electronics. For I/O I planned to use paper tape with a teletypewriter, which could both punch and read paper tape and print as well.

I determined that I could use floating point exclusively if I had a way to deal with the transfer of word segments from one word to another! The 3-address operation that I came up with was Extract, which took "n" bits, beginning at bit "j" in word 1, was to be inserted, beginning at bit "k" in word 2, the result to be stored in location 3. This eliminated the need for approximately a dozen instructions in fixed point! The complete instruction set consisted of 10 instructions – Add, Subtract, Multiply, Divide, Compare (and transfer if the difference is zero or negative), Transfer, Extract, Read-in, Read-out, and Halt. Read-in and Read-out were also very different from any I/O operations I observed for several years following this, until I planned the design of my second computer at IBM in 1955, the IBM 709, when I introduced the I/O channel. Read-in and Read-out specified the information source or sink, beginning at a specified point in the source or sink and beginning at a specified location in the drum storage and continuing until completing the final specified location. The Read-in and Read-out instructions were executed concurrently and independently of computational operations. This overlap of I/O with computing was a major contributor to performance enhancement!

The magnetic drum had sufficient capacity to provide 32 tracks of storage, each containing 32 words of 50 information bits and a 5 bit-length space for track switching time, for a total track length of 1760 bit times. The 50 bit word was made up of 40 bits of numer-

ic fraction, 8 bits of exponent plus 1 bit for exponent sign and 1 bit for sign of the fraction. The arithmetic was performed on the numeric fractions by re-circulating the fractions in re-circulating registers while the exponents and signs were retained in electronic registers for control purposes. The re-circulating registers had the read and write heads spaced 44 bits apart, 40 bits for the fraction and 4 bits for switching time. With this spacing the fraction would have 40 repetitions in a drum revolution, matching precisely the 1,760 bit times in a drum revolution. Each of the arithmetic operations was performed in the course of one drum revolution. I thought I had invented a new way of performing division in one revolution, considering the numerator fraction to be the initial value of the remainder, subtracting the denominator fraction from the remainder and adding a 1 in the leftmost quotient digit position, then shifting the denominator fraction one bit position to the right, preparing for repetition. If at any stage of repetition the remainder became negative the denominator fraction would be added to that remainder instead of subtracted and a 1 would be subtracted in the corresponding quotient position rather than added. I later heard that Dr. John von Neuman had patented it.

Each arithmetic operation and others took one drum revolution to be certain the instruction calling for it to be acquired, then a second revolution to be certain the operands were acquired, then a revolution to perform the operation, and finally a revolution to be certain the result had been stored. Since the operations were non-conflicting, there were four instructions in the pipeline at all times, one picking up its instruction, one picking up its operands, one performing its operation and one storing its result. Consequently the computer performed one floating point operation per drum revolution. I believe there were several world firsts in that design, the first electronic computer to have floating point arithmetic (and certainly the first to have only floating point arithmetic), the first electronic computer to have pipelining, and the first electronic computer to have input and output operated concurrently and independently of computing!

I told one of my fellow physics students about my computer design ideas, and he apparently was excited enough to pass the information on to the Electrical Engineering Department, and in the late fall of 1950 I was requested by them to give a lecture on my design ideas! I gave a seminar and about a week later the head of Electrical Engineering, Dr. Peterson, called my major professor and asked him to change the subject of my doctoral thesis to be a record of my computer design plan so that their graduate engineers could build it and be trained in this new field! My major professor agreed, and I spent six months writing the new thesis and ordering the magnetic drum. I submitted my thesis in June, 1951, expecting to graduate in June. But there was no one at the University who felt competent to properly evaluate it, so it was sent to scientists at the Aberdeen Proving Ground for evaluation. They approved, and I graduated the following February. The thesis was titled "The Logical Design of an Intermediate Speed Digital Computer"; I named the computer the WISC (Wisconsin Integrally Synchronized Computer). It was completed in 1955 and is now displayed in the Computer History Museum in Mountain View, California.

A copy of the thesis was apparently obtained by the IBM branch manager in Milwaukee and sent to IBM at Poughkeepsie. Nathaniel Rochester read it and had IBM make me an offer to join them in Poughkeepsie. I accepted and joined IBM in June 1952. My initial assignment was to simulate neural networks on the IBM 701, according to the proposed characteristics in a monograph published by Professor Hebb. I worked on it for several months and concluded that the description was inadequate. I then turned my attention to character recognition and had considerable success, even on the crude characters of wire printing. The 701 had exhausted its market after the sale of 18 computers. The company decided that a follow-on computer, the 704, should be developed, utilizing the new magnetic core memory rather than the cathode-ray tube memory in the 701, for the capacity could be much larger. I was given the task of designing it, for the other experienced IBM designers were about to be committed to a joint development project with MIT to develop and produce the SAGE system. I decided to double the instruction size in order to accommodate a larger address and additional instructions to provide floating point arithmetic as well as the fixed point arithmetic of the 701. I had heard of an English computer having a "B-box", a counter which allowed the repetition of a loop until the count reduced to zero. Any address step-changing in an array for each iteration still required separate instructions. I thought it would be more efficient if the count and the step-size could be combined, then the program could be shorter and faster. I called it indexing and put three index registers in the 704 to accommodate different step sizes for different data arrays. I assigned two bits in the instruction to identify no indexing and which of the three index registers to use in this instruction. I also discovered that index register contents could be available early enough to modify the address in this instruction before fetching the data, thus having no additional execution time! It turned out that the Sage system also had an indexing capability, but I don't know who had it first; they were a classified project, and I wasn't cleared, so I had no knowledge of it to get the dates of invention.

When the time came to price the 704 for the market, it was necessary to estimate the probable market size. Pricing people from IBM headquarters came to talk to me and get my agreement on size. They initially estimated a market of six machines (I assume they considered the 18 701 machines had mostly satisfied the market). I was

incensed and insisted that the machine had so much more capability than the 701 that it would have a larger market size. Over the next few weeks they came back with 12, then 18 and finally 32 before I agreed. The actual number sold was 140, making it an extremely profitable program!

I was then asked to design the follow-on system, the 709, and shortly thereafter to design a supercomputer (called STRETCH) to utilize the new technology, transistors. I was told it would be my project, but that I would have to get a development contract, preferably from the Livermore Labs or Los Alamos. This was late November 1954. I consulted a bit with John Backus, and we agreed on the principal characteristics the STRETCH should possess. I studied the capabilities of the proposed semiconductor technology, which was a new circuit type called ECL (Emitter Coupled Logic), sort of like the vacuum tube push-pull amplifiers. They were extremely fast circuits and very power consuming. I did some designing of a multiplier to estimate the probable performance that could be achieved if efficiently instructed. I then worked on a new concept "look ahead" which consisted of fetching instructions well in advance of their execution time so that branch instructions could be recognized early enough to fetch an alternative instruction sequence with no delay. The design analysis was very promising, yielding several times the performance we could achieve using vacuum tubes. Armed with my initial design results I visited Livermore first. They listened and were very cordial, but they informed me that they had already committed to contract with a competitor so they couldn't commit to us. I then visited Los Alamos and presented to them; they were very interested and would negotiate with IBM. I then did some more fleshing out of the STRETCH design and also determined what should be done to the 704 to produce a 709. In the 709 project I added a number of reasonably useful new instructions, one of the most interesting ones was a "history dependent table look-up", which allowed code conversions from BCD (Binary Coded Decimal), IBM's preferred code, to ASCII, the newly adopted American Standard Code for Information Interchange, or vice versa. It also allowed two binary coded decimal numbers (each decimal digit occupying a 6-bit character position) to be added or subtracted in binary, then using the table look-up to convert the result to a proper binary coded decimal result. These were two examples, but many more were ultimately developed by customers. The principal change I wanted to make was the introduction of an I/O channel, permitting the computer to specify the reading or writing of a specified amount of data to or from a magnetic tape or drum into or out of memory without the computer having to control the data flow as it occurred, just as I did in the WISC, but be able to continue computing with only the impact of some memory cycle delays due to conflict of memory requests. This change was a significantly costly development so that it required corporate approval. Elaine Boehm and I determined that we had to make an outstanding demonstration to win approval. We came up with the idea of a tape sorting program. The IBM 703 was a sorter-collator, a fairly modestly priced machine sold to the US Treasury. The expected price for the 709 we estimated to be at least two or three times that of the 703. We programmed the sort and found that it performed so much faster than the 703 that the cost of sorting on the 709 was less than on the 703. This demonstration tipped the balance and the I/O Channel development was approved! At this time (mid 1955) I was surprised to have a man assigned to my STRETCH project. I initially assumed he reported to me, but it became clear that he thought I reported to him. This was very disconcerting, for I had been assured that STRETCH was my project before I accepted the assignment and had then gotten Los Alamos to the negotiating table and had achieved quite a bit of the design. I wasn't certain I had the situation figured out for sure, so I continued on. This new man was uninterested in my design and had his own approach. He wanted to design a front end computer which would be a commercial computer which would then feed the back end which would be the scientific computer. To me this seemed to totally prevent any possibility of resulting in the supercomputer that I was commissioned to design! Late that year I was invited to meet with the Laboratory manager; he showed me his plan for restructuring the Laboratory. It was to be a matrix structure with several development projects feeding the technology engineering groups. The STRETCH development was to be managed by the man assigned to me a few months earlier. I was to be in charge of the STRETCH detailed design. I was appalled, for I knew we could never agree, and the project would fail. I didn't respond about my reaction; I just went back to my office and wrote my letter of resignation. I did continue on until just before Christmas, providing my best design ideas, all of which were lost, then left for South Dakota for Christmas with my and my wife's families, then on to Los Angeles to join Ramo-Wooldridge's computer division.

Almost five years later Dr. Piore, IBM's Chief Scientist who reported directly to Tom Watson, came to Los Angeles and invited my wife and me to have dinner at Romanoff's restaurant in Beverly Hills. Dr. Piore's wife was from the Romanoff family, so we had remarkable attention! Dr. Piore offered me the position of managing the Experimental Machines division in IBM Research, with the requirement to be on the East coast for a minimum of four months and a maximum of seven months. My wife and I accepted and were back in New York State by November 1960. My first activities were to look at the projects in my division. I cancelled the only two hardware design projects because they had no chance of being of value to IBM. One project was a computer design which

had been continually changed but never complete enough to be evaluated; the other was a government project which utilized super conductor switches for logic, but there was no way to amplify diminishing signal levels. This left only software projects, which I retained, and the responsibility for designing a new supercomputer with insignificant funding. I believe I was given this because the STRETCH project had not met its performance target so its price had been reduced and became a loss leader. The 704, 709, and the follow-on 7090 and 7094 had sustained IBM's scientific computing market.

At this time IBM had the SPREAD committee in session. There were about five major computer families made by various IBM divisions, each of which had generations which weren't quite compatible. Unfortunately, the total development costs were growing impossibly large, for any new device to be attached required an engineering and software project to be manned and funded for each member of each family. So IBM's development budget was greater than most computer companies' revenue! The SPREAD committee's goal was to define data formats, kinds of I/O devices, control, storage and logic technologies which were to be standardized, and to plan a new family of computers which would replace all current families. This was not only an enormous undertaking, but it was even more of a political undertaking, for it would require all of the divisions to yield their fiefdoms to the king! This was a major revolution being fought, but the stakes were high, managing the costs to maintain control of the world's data processing market-place! I had only been back east for a couple of months when I was approached by the president of the Data Systems Division, Bob Evans. He asked me to meet with him at a budgeting session to be held at a small resort called Jug-End. I sat through a session that amply demonstrated the development budgeting problem. After that session Bob and I met privately, and he asked me if I would consider designing the new family of computers. I asked him if the new family of computers would be upwardly compatible but not downwardly compatible. He said that was the plan. I said I would not be willing to do that, for it would only end up with the same budget problem I had just observed, for the generational problem would exist immediately. I told him I thought the family could be both upwardly and downwardly compatible and with virtually no cost impact, and if they would enforce this constraint I would be willing to accept the challenge. Bob thought for a moment, presuming I could possibly do it, and he agreed. So in 1961 I was moving back to Poughkeepsie, where I worked for about 10 hours a day defining data formats, the instruction set, and in several cases the hardware structure, for each family member was to have about a factor of three difference in performance from its neighboring members which required registers in the smaller machines to be memory locations, but in larger machines to be in circuitry. There were a total of 7 machines in this System 360 family, covering a performance range of about 600 to 1. It is still IBM's mainframe line, though changed through the decades, and is IBM's largest revenue product, and as described in Halloween's Palo Alto Daily News, it is superior to complexes of minicomputers or PC's!

To meet the performance and cost constraints, the small machines had to use memory locations as registers in appropriate cases, where the larger machines could use hardware registers. I also discovered that there had to be some portion of the architecture that had to be reminiscent of each of the two significant families that we were replacing, otherwise the designers from those families couldn't develop the confidence that the design would be acceptable in their market segment. This resulted in decimal operations being memory to memory rather than in registers like the 1401 and indexing very similar to the 7094. There were only two architectural advances of note in the full family; the most significant was "base registers," which allowed a much smaller address size required in the instruction format to address a quite larger memory (I believe the inventor was Dr. Gerritt Blau), and the other was making the addressing of the disk storage and tapes to be sufficiently alike so that the users familiar with tapes could experiment with the new disk storage without having to use random access exclusively (this was largely due to 1401 I/O designers I believe), in the fastest member, the model 90's there were three very powerful ones, loop trapping, associated with look-ahead, planned by Dr. Tien Chi Chen and myself, then virtual registers (registers assigned when and where needed) and linked arithmetic units, so results of one arithmetic unit could become an input to another arithmetic unit without any intervening register storage; these were planned by the regular design team. The principal negative consequence of the SPREAD committee data format constraints appeared in floating point, where having to use binary sizes for the exponent size, eight bits, and for fraction shifting by multiples of four bits, for the rounding errors were larger than I thought reasonable. I tried to get relief from the constraint in this case, but was refused. It took about 20 years before IBM switched to IEEE floating point format. In the mid 1970s, the architectural concept invented in England called "Virtual Memory" was introduced into the IBM System 360, and with some months to learn how to use it efficiently, became a very effective advance! Amdahl computers during that time utilized IBM instruction sets that could employ the IBM operating system, which was almost universal in the computing market-place. Consequently, Amdahls didn't contain architectural advances which altered instruction results, but did contain pipelining as in the WISC and had much more advanced technology, such as LSI (Large Scale Integration)

Dr. Amdahl holding a 100gate LSI air-cooled chip. On his desk is a circuit board with the chips on it. This circuit board was for an Amdahl 470 V/6 (photograph dated March 1973).

Photograph of the Amdahl 470 V/6.

with air cooling, a world first (developed by Fred Buelow), rather than IBM's MSI (Medium Scale Integration) with water cooling. Amdahl also included another world first, remote diagnostics, called "Amdac," invented by the field engineers.

During this time, I had access to tape storage programs and data history for commercial, scientific, engineering and university computing centers for the 704 through the 7094. This gave insight on relative usage of the various instructions and a most interesting statistic -- each of these computing center work load histories showed that there was 1 bit of I/O for each instruction executed! I also was able to determine the speed of computing that could be maintained for a given memory size. This was related to disk and tape speeds in the environment of multi-processing. These latter two properties I determined in 1969, when I privately estimated that System 360 would have to change the address length to exceed about 15 MIPS (Million Instructions Per Second). Livermore Laboratory heard about the 1 bit of I/O and thought it couldn't be true so they ran tests for a month and found that during office hours, when users were using the machines from their consoles, the number of bits of I/O averaged 1.1, and at night doing batch processing it averaged 1.0. They were surprised, but neither they nor I knew why it should have that value. When virtual memory came into common usage, the number of bits of I/O per instruction executed came down. Although I had limited data, I could reasonably estimate that it correlated quite closely with the reduction of the percentage of the program size which hadn't needed to be brought in or retained during the course of its execution.

In 1967 I was asked by IBM to give a talk at the Spring Joint Computer Conference to be held on the east coast. The purpose was for me to compare the computing potential of a super uniprocessor to that of a unique quasi-parallel computer, the Illiac IV, proposed by a Mr. Slotnik.

The proposed Illiac IV had a single Instruction unit (I-unit) driving 16 arithmetic units (E-units). Each E-unit provided its own data addresses and determined whether or not to participate in the execution of the I-unit's current instruction, an interesting, but controversial proposal. The super uniprocessor was a design type, not a specific machine, so I had to estimate to the best of my ability what performance could reasonably be achieved by such a design. Figure 1 shows a diagram of the Illiac IV; Figure 2 shows the performance of the Illiac IV on a problem having a varied, but reasonable, range of parallelism under the control of an operating system with characteristics similar to those then currently in use, having quite a bit of system management and data movement code.

Figure 3 shows the performance of the super uniprocessor on that same problem and operating system; Figure 4 shows the performance of the Illiac IV with Slotnick's expected future goal of 256 E-units and running a problem having a varied range of parallelism, but reaching a level of parallelism matching "America's symbol of purity," Ivory soap, 99.44%.

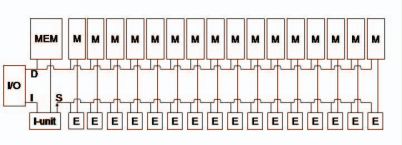

Figure 1. A diagram of the Illiac IV.

Figure 5 shows the formula I generated to estimate the Illiac IV's performance, giving it the benefit of assuming that if some parallelism existed all processors could be usefully employed.

These Figures are not quite the same as in the 1967 presentation, for they weren't published, nor did I keep them, for I had no expectation of the intensity of their afterlife! I never called this formula "Amdahl's Law" nor did I hear it called that for several years; I merely considered it an upper limit performance for a

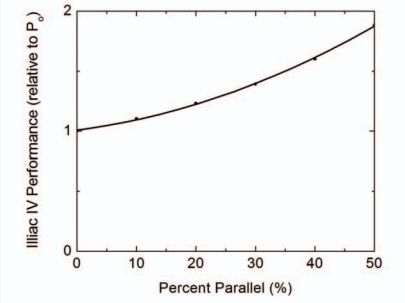

Figure 2. Illiac IV Performance

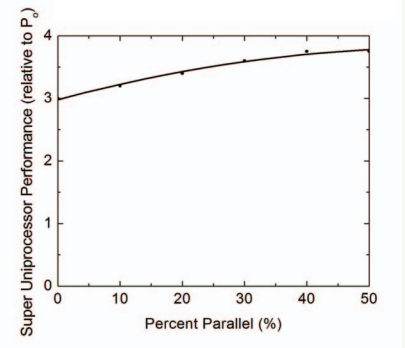

Figure 3. Super Uniprocessor Performance.

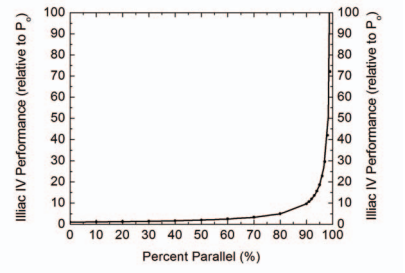

Figure 4. Performance of the Illiac IV with Slotnick's expected future goal of 256 E-units.

computer with ONE I-unit and N E-units running problems under the control of that time period's operating system! The debate between me and Mr. Slotnick was joined by many in the audience, and it became quite heated. I felt Mr. Slotnick was trying to egg me into attacking him rather than his computer design, but I carefully avoided that, only to be attacked by Dr. Herbert Grosch in the audience for not attacking him. It became a bit of a circus, and I was quite unhappy about being involved, for I thought of it as a rational analysis of two competing design approaches, not a bashing of another human for offering a controversial design approach! Several years later I was informed of a proof that Amdahl's Law was invalidated by someone at Los Alamos, where a number of computers interconnected as an N-cube by communication lines, but with each computer also connected to I/O devices for loading the operating system, initial data, and results. This made all control and data movement to be carried out in parallel. I didn't enter the fray; I merely commented that what they called Amdahl's Law merely described the Illiac IV, which had only one I-unit. I also heard that Amdahl's Law was used to challenge the multi-computer systems developed by Massively Parallel, a Colorado firm, where their chief system designer looked at the formula and though it appeared to have the form of an information theoretical statement and used it to further enhance his system! As a result Massively Parallel invited me to join their advisory board! I really still do not consider Amdahl's Law to be as much of a law as the relationships of memory size and computer performance and also the number of bits of I/O per instruction executed or as reduced, when considered as a function of the fraction of program required to be in "virtual memory". These seem to be lost in the mist of time! I also consider the WISC to be the most remarkable architectural achievement I've made, and with no input from any source other than sheer inventiveness.

$$P_p = P_s\left(\frac{S+P}{S+\frac{P}{n}}\right)$$

Figure 5.

The formula generated to estimate the Illiac IV's performance. The numerator in the formula is Ps x (S+P), and the dominator is S+P/16. In this formula, S is the % of the problem that must be executed sequentially (or serially), and P is the % of the problem which may be executed in parallel if the computer has this capability. The sum (S+P) is always equal to 100%, or 1, for it is the workload to be performed. Ps is the performance of a computer which can only execute the problem in a totally sequential manner, regardless of the problem possessing the capability for parallel execution, and which has the speed of the Illiac IV's I-unit. The denominator reflects the capability of the Illiac IV to be able to execute the P component 16 each instruction execution time (this is giving an advantage to the Illiac IV, for only in vector or matrix operations where the sizes are multiples of 16 would it perform that well). The ratio (S+P)/(S+P/16) represents the speed-up of the Illiac IV architectural feature for parallelism. This speed-up times Ps is the optimistic performance of the Illiac IV. By the way, no one challenged this formula, just my range of only up to 50% parallel was thought by some to be a bit low.

There has been no publicity about the capability of the actual Illiac IV. I did hear unofficially that it was unable to be successfully debugged at the University of Illinois and that it was shipped to the NASA facility here in Sunnyvale where debugging was being carried out by volunteers. I heard a few months later that they had gotten it to work and had executed a test program, but that no information on its performance had been made available. I'm not certain that this information was entirely accurate so I cannot vouch for it.

From: Naintre, Paul
Sent: Friday, March 30, 2007 11:29 AM
Subject: Arrivederci

Hi Folks,

The day has arrived after 27 years with Amdahl, when it's my turn to say goodbye.

It was a pleasure working with such great people as yourselves.

I always likened Gene Amdahl to Enzo Ferrari: both builders of powerful, gleaming, machines.

Enzo's machines were measured in horse power, max rpms, engine size and number of Formula 1 wins.

Gene Amdahl's machines were measured in MIPs, max transactions per second, no. of CPU engines and number of customer wins.

Two completely different industries, both built from passion, courage and a desire to produce the world's most powerful, revered, machines.

We certainly made the best products in the world and I was very proud to have the opportunity to work here with such great people.

I wish you all Good Luck and all the very best for the future.

Paul.

p.s. If you would like to stay in touch my new email id is: paul.naintre@gmail.com.