

Wie schon eben gesagt, ist es schon wichtig, wenn man bei so einem Forschungsgebiet schaut, wie sich das entwickelt hat, was sind Milestones und vor allen Dingen Lessons learned. und lernt auch aus seinen Fehlern. Und das wollen wir uns in diesem Kapitel ein bisschen anschauen. Um dieses Kapitel zu verstehen, müssen Sie so eine Grundahnung haben von Rechnerarchitektur, entweder aus der Vorlesung Grundlagen der Rechnerarchitektur und oder aus den Rechnerstrukturen, was wir bisher gemacht haben. Sie sollten das Wort Instruction Level Parallelismus verstehen, also wie viel Parallelität steckt in einem Befehlssatz. Und worum geht es in dem Kapitel? Parallelrechner gebaut wurden und was kann ich eben aus dieser Geschichte lernen. Wir werden uns also im Inhalt so die wichtigsten Hochleistungsrechner am Beispiel mal anschauen und was High Performance Computing betrifft, die Figur oder die Person, die das am besten vertritt, war Seymour Cray mit seinen Rechnern. Die Firma Cray gibt es ja auch heute noch und da wollen wir uns natürlich anschauen, Parallelrechner anschauen, also die ersten SIMT-Ansätze mit Feldrechnern und dann so dieser grundsätzliche Wettstreit, ist jetzt ein Multiprozessor mit gemeinsamen Speicher besser oder ein Multiprozessor mit verteiltem Speicher, da hat es jeweils Realisierungen gegeben und dann schließlich noch einen kurzen Blick auf neue Parallelrechnerwaffen. Und da gibt es natürlich die Top 500, das heißt die 500 schnellsten Rechner weltweit an der Spitze, welche Architektur ist denn derzeit die beste, die schnellste. Ja, also im positiven Sinne, ein Verrückter war der Herr Seymour Cray, der hat gearbeitet bei der Firma CDC Data und die Rechner, die die gebaut haben, das waren die Cyber 6600, Cyber 7600, das waren also die für die damaligen Zeit rechner die waren natürlich entsprechend teuer also nichts was man sich so im wohnzimmer hinstellen kann sondern ganze turnhallen voll gerätschaft mit kleinem kraftwerk nebendran und millionen dollar schwer natürlich im jahr 64 gab es die cdc 6600 mit einer leistung von 5 bis 9 pro Sekunde. Wenn ich diese Zahlen immer sehe und dann überlege, dass die erste Mondlandung 1969 war, dann wundert mich das schon ein bisschen. Natürlich hat die NASA ein, zwei solcher Rechner gehabt, aber dass man mit dieser doch relativ geringen Leistung von 40 Megaflop pro Sekunde es geschafft hat, erfolgreich auf dem Mond zu landen, das ist also schon eine Top-Leistung IBM 360 war ja der Standard-Bürorechner für die Industrie, also kein Homecomputer natürlich für die Industrie im Jahr 1969. Der hat eine Leistung von 330 Kiloflops gehabt. Also die anderen Rechner waren da ungefähr 120 mal so schnell, die Hochleistungsrechner der damaligen Zeit. Man hat diese Rechner auch als Supercomputer bezeichnet. Und Supercomputer heißt einfach, Kosten spielen keine Rolle, Energieverbrauch spielt keine Rolle, Hauptsache das Teil ist schnell. von der Formel 1 der Rechnerindustrie. Man hat damals schon Superskalarität implementiert in diesen Rechnern. Superskalarität, wie wir sie besprochen haben, ist keine Erfindung der 2000er Jahre, sondern die stammt aus den 60er Jahren. Man hat als Schaltungstechnik schon Transistoren gehabt, aber keine ICs. Das waren Boards mit aufgebrauchten Transistoren. Dementsprechend Da sie groß waren, haben die Laufzeiten auch eine Rolle gespielt. Deshalb hat zum Beispiel die Cyber eine Y-Form gehabt, um eben die Verbindungskabel zwischen den einzelnen Boards, die Entfernungen möglichst kurz zu halten, um

die Maschine möglichst schnell takten zu können. Speicher, damals Rinkernspeicher, also Verrittkernspeicher. aufgezogen, also mit Schreibtraht, Lesetraht. Das heißt, man konnte jedes Speicherbit und jeden Transistor in so einer Maschine noch mit bloßem Auge sehen. Und wenn man sich dann so eine Turnhalle voll Gerätschaften mit Kraftwerk anschaut, dann hatte die eine Rechenleistung, die in etwa von so einem Pentium mit 200 MHz auch abgeliefert wurde. Und dann schaut man noch mal drauf, okay, dieser Rechner wurde ungefähr 1970 gebaut, Ahnung, in den 90er Jahren, Ende der 90er, Mitte der 90er Jahre, also 25 Jahre später hat man das Ganze on Chip bekommen und das war das Dilemma der Parallelrechner damals, weil einfach diese Mikroprozessoren so wahnsinnig schnell geworden sind. Gut, ja, Superskalarität in der Cyber, wir sehen hier welche Einheiten da waren, Funktionseinheiten, einen Shift-Prozessor, Floating-Point-Verarbeitung, Fix-Point-Verarbeitung, zwei Multiplikationseinheiten, eine Divisionseinheit und dann extra für die Speicheradressberechnung nochmal Einheiten. Es war ein 10 MHz Takt der Maschine. Ziel war auch, dass man pro Takt einen Befehl ausführt, also CPI, Cycles Construction von 1. Man hat wir jetzt kürzlich besprochen haben in der Vorlesung, um eben mögliche Konflikte zu lösen. Es gab I.O. Peripherieprozessoren und was damals natürlich schon so war, dass jedes einzelne Bit, um jedes einzelne Bit wurde gekämpft, weil jedes einzelne Bit war teuer. Das ist ja heutzutage nicht mehr so ganz der Fall und dementsprechend gab es da durchaus auch Architekturen, die eben keine Zweierpotenzen waren. 60 Bit und ein Maschinenwort war eben auch 60 Bit breit. Ich habe ein einheitliches Befehlsformat gehabt, also das war damals eigentlich schon eine Risk-Architektur, wenn man das mit heutiger Terminologie bezeichnen will. Und so ein 60 Bit Wort konnte ich natürlich unterschiedlich aufteilen. Entweder stand da drin so ein 64 Bit Fließkommabed, also heute würde man sagen ein Doppel mit IEEE, das sind natürlich 64 Bit, aber damals war es halt nur 60 Bit, also ein bisschen weniger Darstellungsgenauigkeit. Oder ich habe 15-Bit-Register gehabt, 60 durch 15 gibt 4 Register. Oder ich habe 30-Bit-Memory-Adressen gehabt, also 2 mal 30, konnte also 2 Adressen in so einem Wort unterbringen. Oder ich habe meine Zeichen, die ich da verwendet habe, mit 6 Bits kodiert, konnte also 10 Zeichen in so einem Wort unterbringen. Also man hat damals auf Groß-Klein-Schreibung natürlich verzichtet und Sonderzeichen auch weitestgehend, ausgereicht haben um alle Zeichen die ich verwendet habe zu kodieren heutzutage kann man ja klar Byte also 8 Bit pro Zeichen eine 8 Bit Kodierung allerdings die eingesetzten Programmiersprachen Vortrag Kobol da war es durchaus üblich dass man eben keine Großkleinschreibung beachtete das heißt alles waren Großbuchstaben und man auch keine Sonderzeichen brauchte Also für Registeroperationen habe ich 12-Bit-Instruktionsbreite und für Speicheroperationen habe ich 24-Bit-Instruktionsbreite, wobei eben 18-Bits für die Adresse hinfielen. So, jetzt kann man jetzt wieder ausrechnen, 2^{18} hoch 18 gibt also 256-Kilo-Worte, Worte, nicht Bytes. Ja, ein Wort ist 60-Bit, die ich da adressieren konnte. ganz gestimmt hat, der hat nämlich auch gemeint, kein Mensch braucht mehr als 250 Kilowatt Speicher, das ist wahnsinnig viel. War es auch für die damalige Zeit, aber natürlich braucht man mehr Speicher, wie die Zukunft dann gezeigt hat. Warum wollte er nicht mehr Speicher, weil der

Speicher war natürlich irre teuer. Wir reden von Trinkhandspeicher, 24 Bits mal 12 Bits abspeichern, aber das war dann schon ein Kuchenblech praktisch. Und 5 Module gibt dann eben 1024 Speicherworte von 60 Bit. Und 5 Module, da brauchte ich schon einen Hubwagen, um die durch die Gegend zu fahren. Und wenn ich dann eben so einen 256 KW Speicher voll aufgebaut habe, also 256 diese Module brauchte, da war wahrscheinlich ein 40 Tonner schon ganz gut beladen mit dem Speicher. obwohl der Speicher so teuer war, haben die Kunden relativ schnell mit dieser Maschine, die mit dieser Maschine gearbeitet haben, noch mehr Speicher verlangt. Also zum Beispiel so eine Firma wie Boeing, die hat dann 2 Megawatt zusätzlichen Speicher verlangt. Den hat man dann auch gebaut, da muss man natürlich so wieder eine Schnittstelle bauen, die war ein bisschen langsamer, also 1,4 Mikrosekunden Zugriffszeit, hatte, also ungefähr dreimal so schnell war. Aber auch hier kam schon wieder das Prinzip der Parallelität zum Einsatz. Ich konnte über einen 600-Bit-Bus 10 Worte gleichzeitig lesen, während der Hauptspeicher eben nur mit diesem 60-Bit-Bus direkt verbunden war. Also durch Parallelität konnte man diesen Geschwindigkeitsnachteil dann wieder ausgleichen. Zeit bestreben war, immer die Maschine schneller, schneller, schneller zu machen. Und wir erinnern uns an die Performance-Formel, eine einfache Möglichkeit, die Maschine schneller zu machen, ist, die Taktzeit zu verkürzen. Und man sieht es schon, von der 6600 zu 7600 Cyber hat man eben die Taktzeit um ein Viertel, also viermal so schnell gemacht, das heißt von 100 Nanosekunden auf 27,5 Nanosekunden gedruckt. unter anderem unter Architektur ist man eben auf diese 40 Megaflops gekommen, die für die damalige Zeit natürlich wahnsinnig viel waren. Gut, die nächste Idee, Architekturidee, die Seymour Cray dann auf den Markt gebracht hat, war die der Vektorisierung. rücksichtslos und skrupellos war, was Kosten betroffen hat für seine Ideen. Und die Firma Control Data Corporation natürlich auch an den kommerziellen Erfolg ihrer Rechner dachte. Und im Streit ist dann Simon Cray von der Firma CDC weggegangen und hat seine eigene Firma gegründet. Und sein erster Rechner war die Cray 1S. Und was neu war, war eben das Prinzip der Vektorverarbeitung. Man könnte auch ein bisschen Pipelining, Auf jeden Fall war diese Maschine darauf spezialisiert, Vektor-Operationen sehr schnell ausführen zu können, weil einfach eine Beobachtung war, dass viele Vektor-Matrix-Berechnungen durchgeführt werden und es an der Stelle was bringen könnte, wenn man diese Operationen beschleunigt. Man hat also Vektorregister eingeführt, 64-Bit-Architektur und hat das Ganze gepipelined, das heißt pro Takt ein Ergebnis, auch wenn ich jetzt komplexere Operationen wie Multiplikation war. Man hat also einen ganzen Satz von Registern gehabt, einmal Vektorregister, dann Skalaregister und dann für die Adressrechnung extra Adressregister. Der Speicher entsprechend groß, also bis zu 1 Megawatt ausbaubar, Serienausstattung war 1 Megawatt und der Speicher war auch nochmal gespiegelt, man konnte also schnell zwischen Adressbereichen und Werten umschwenken, einfach natürlich relativ teuer für die damalige Zeit, aber auch entsprechend schnell. Also es gab einige Installationen in Deutschland, an die ich mich noch erinnere. Und hier sieht man mal so ein Grouschaltbild. Man sieht hier die Superskalararchitektur, man sieht aber auch, dass es Vektoreinheiten gibt, also die Vektorisierung, spezielle Register, die Vektoreinheiten,

die also Vektoroperationen besonders schnell machen konnten. Aber darüber hinaus hat man auch schnelle Skalareinheiten. Die Idee war natürlich, wenn ich ein sehr schnelles System haben will, also ein Superrechner, muss alles schnell funktionieren. Sowohl Skalaroperationen als auch Vektoroperationen und das hat man mit dieser Cray 1 erreicht. Gebaut wurde die 1976, Cray hat es geschafft damals schon einen 80 MHz Takt zu fahren, also 12 Nanosekunden. Dann haben wir eine Ausbreitung vom elektrischen Feld von 0,3 Meter pro Nanosekunde. Das heißt, der längste Weg, den ich da verbauen dürfte für die Leitung, war 3,6 Meter. Und wir sprechen ja von Rechnern, die so Räume noch gefüllt haben und da sind natürlich 3,6 Meter wenig. Das heißt, ich muss auch mein Gehäuse so optimieren, dass da keine längeren Verbindungen drin auftauchen. 160 Megaflops, also viermal so schnell als die Vorgängermaschinen. Und besonders erfolgreich war die Architektur, weil eben alles schnell war. Also sowohl die skalare Architektur und auch die Vektorisierung besonders schnell war. Gut, so ein Rechner war schweineteuer und so ein Rechner wurde im Wesentlichen verwendet. Das heißt, ich habe natürlich auf so einem Rechner keine Programmentwicklung gemacht. Das wäre Verschwendung von Ressourcen gewesen. Ich habe also auf einem billigen IBM oder anderen Rechner meine ganze Programmentwicklung gemacht, dann habe ich gross kompiliert und ich habe das Programm nur ausgeführt auf der Cray 1. Also so war früher und ist auch heute teilweise noch die Vorgehensweise, dass dieser teure Supercomputer nicht für normale Verwaltungsaufgaben verschwendet wird, sondern dass es dessen Aufgabe ist, rechenintensive zu erledigen, die laufen dann auch oft noch im Batchbetrieb, das heißt da werden die Jobs einfach drauf geschoben, erledigt und dann kommt der nächste Job und die Vor- und Nachvisualisierung zum Beispiel, die wurde eben auf anderen Rechnern ausgeführt. Also die Cray war einfach zu teuer für normale Operationen, es war ein reiner Rechenknecht, hat man dazu gesagt. 4,1 Nanosekunden Zykluszeit, also 1,2 Meter, das heißt ich musste die Maschine auch physikalisch verkleinern, damit das Ganze funktioniert. Wir haben ein Multiprozessorsystem, wir haben also 488 Megaflops, bei 4 Prozessoren gibt es natürlich mehr als 1 Gigaflop Leistung. Es wurde versucht die Größe natürlich so klein wie möglich zu halten, das heißt 1,35 Meter Höhe, der 15 Durchmesser. Zur damaligen Zeit war es nicht möglich das Ding noch Luft zu kühlen, das heißt es wurde mit Fluor Kohlenstoff flüssig gekühlt. Der Rechner war 2,5 Tonnen schwer und das Kühlmittel davon war ungefähr ein Drittel vom Gewicht. Ich habe eine 32 Bit Adressierung, ich habe 512 Megawatt, das heißt 4 Gigabyte Speicher, was für die damalige es ist ein super Computer, also wir reden von der Formel 1. Cray hat natürlich unterschiedliche Rechner verkauft er hat auch die Cray XMP, Cray YMP auf den Markt gebracht, die waren eher für Vektorisierung ausgelegt und die haben mit mehr Prozessoren dieselbe Rechenleistung erlegt, also die Cray 2 war halt der schnellste General Purpose wenn man so will, Rechner. Gut, außerdem hat er Interconnect angeboten, das heißt, wenn jetzt eine Cray 2 nicht reicht, der konnte die mit einer schnellen Verbindung eben mit anderen, also mit 1,6 Gigabit pro Sekunde vernetzen, was auch für die damalige Zeit exorbitant eben war. erstmal eine neue Schaltkreistechnik entwickeln. Also Silizium hat für diese Taktung nicht genügt, das heißt man hat mit Gallium Arsenid Transistoren experimentiert

und hat ein bisschen gedauert bis man die Technologie im Griff hatte, damit man eben entsprechend mit 2,1 Nanosekunden takten könnte. Wir sind bei knapp einem GigaFlop Prozessorleistung, bei im Jahr 1993 und ja die Maschine war von den Zahlen her toll ja aber kommerziell ein kompletter Fehlschlag weil sich schlicht und ergreifend kein Unternehmen so eine Maschine leisten konnte also wir haben genau 0 verkaufte Systeme weil erstens mal war die Anschaffung zu teuer und zweitens der Unterhalt exorbitant. Das wollte sich also niemand ans Bein binden. Ja, das hatte Herrn Cray aber nicht besonders interessiert, sondern er war ausschließlich an Leistung interessiert. Das heißt, er hat schon an der nächsten Maschine gearbeitet, an der Cray 4. Zykluszeit und Ziel war eine Nanosekunde, Technologie wie der Gallium Arsenid. Und leider ist der Herr Cray dann, er war wohl auch im Leben schnell unterwegs, an den Folgen eines Verkehrsunfalls gestorben, hat sich dann die Firma ein bisschen konsolidiert. Das heißt, es hat immer noch schnelle Prozessoren gebaut und auch verkauft, aber eben nicht mehr in dem Maße Technologie getrieben, wie Cray das gemacht hat. Gut, das war also die Geschichte der Firma Cray. Die gibt es ja auch heute noch. Die bauen eben, wenn man so will, Spezialprozessoren, schnelle Prozessoren, immer noch teuer und leistungsstark. Projekte, also Cray hat halt die Technologie an die Spitze getrieben, also sein Ziel war immer, möglichst schnell takten, kostete es, was es wollte. Auch wenn ich dann eben kein normales Silizium verwenden kann, sondern Gallium Arsenid verwenden muss, kein Thema, dann machen wir das einfach. Und die Grundidee war, Matrix-Operationen zu beschleunigen und zwar eben mit der SIMT-Technologie. Also SIMT steht für Single Instruction Multiple Data. Das heißt, wir haben einfach festgestellt, dass wir bei großen Matrizen für jedes Element der Matrix immer wieder die gleiche Operation ausführen. Also brauche ich einfach nur einmal die Operation aussprechen und mache dann diese gleiche Operation auf möglichst vielen Elementen gleichzeitig. Stammen aus dem Bereich der Lösung von partiellen Differenzialgleichungssystemen also pdgl partielle Differenzialgleichungen und ja das ist zum Beispiel bei der NASA mit ihren CFD Problemen also CFD steht für computational fluid dynamics also irgendwelche Strömungsmechanik Probleme die da berechnet wurden so was man jetzt natürlich gemacht hat dass man seine Hardware so ein bisschen anders gemacht hat, das heißt, wenn Sie sich so eine Matrix anschauen, die erinnert ja sehr stark an ein zweidimensionales Gitter und dementsprechend hat man seine Hardwarestruktur auch nachempfunden, hat da so ein zweidimensionales Gitter aufgebaut und die Globalarchitektur war, dass man also 64 Prozessoren parallel geschaltet hat, also Processing Units und so eine Processing Unit besteht aus einem Processing Element und einem Processing Memory Über diesen Processing Units gab es eine Control Unit, also einen Host. Das war der Supervisor, das heißt, der hat die Operation vorgegeben und die anderen haben diese Operation dann ausgeführt. Es gab zwei Verbindungsnetzwerke, einmal für die Control Unit zu den Processing Units. Das war eine zentrale Steuerung, also so eine Sternstruktur. Alle Processing Units waren direkt mit der Control Unit verbunden. Und dann waren die Processing Units noch benachbart verbunden im zweidimensionalen Gitter. Das heißt, jede Processing Unit hat vier Nachbarn, Nord, Süd, Ost, West. Die einzelnen Knoten haben eine 64-Bit-Verarbeitung

gemacht. Die konnten schnell multiplizieren und addieren und einfach so einen normalen Befehlssatz für so ein Processing Element. Auch hier hat man wieder so Milestones in der Rechner Architekturentwicklung gemacht. Man hat also das erste Mal schnelle ICs eingesetzt, also mit der Coupled Logic, als die normale MOSFET Logik, allerdings natürlich auch ein bisschen mehr Leistungsaufnahme hat, also ein bisschen wärmer wird das Ganze, auch fertigungstechnisch ein bisschen schwieriger zu beherrschen, aber eben für Hochleistungsrechner, da spielen ja die Kosten nicht so die ganz große Rolle. Dann hat man natürlich das erste Mal Halbleiterspeicher im Großeinsatz gehabt, also auch hier war so der Übergang auf integrierte Schaltkreise zu sehen, gesammelt im Bau von multi layer bots auch das war jetzt das erste mal was muss man mitgearbeitet hat ja das ist das meine ich mit warum soll man sich geschichte anschauen weil dann sieht man auch wo das ganze entstanden ist wie es entstanden ist und solche der Bau von solchen super rechnern war gesagt wie gesagt wie in der formel 1 also es gibt auch bestimmte sagen so überflüssig man die formel 1 im autobau betrachten mag so gibt es doch das eine oder andere technologische auch Eingang gefunden hat in normale Rechner und so ist es hier natürlich auch. Ja, ok, so, das war jetzt dann die wesentliche Architektur, also das sind die Processing-Elements mit dem Speicher und darüber angesiedelt war natürlich das Supervisory Computer System, also die Control-Unit, ja und wenn man sich das jetzt so einfach darstellen will, dann hat diese Control-Unit jetzt gesagt, ok, wir haben jetzt eine Matrix ihr macht jetzt einfach eine Addition von zwei Matrix-Elementen, jeder auf unterschiedlichen Daten. Und dann haben die 64 Processing-Elements, die haben nur zwei Möglichkeiten gehabt, entweder sie machen diese Addition oder sie machen sie eben nicht diese Addition. Mehr Möglichkeiten gab es nicht, also die konnten nicht dann irgendwas anderes machen an der Stelle, sondern das war alles global synchron getaktet und sie haben nur die Möglichkeit gehabt, an dem Befehl teilzunehmen Arbeitsweise. Single Instruction Multiple Data. Dafür war diese Maschine optimiert und diese Maschine war natürlich dann auch relativ schnell, wenn die Anwendungen dazu gepasst haben. Man kann sich gut vorstellen, dass für Matrix-Operationen sowas natürlich vorteilhaft ist, so eine Architektur, aber eben weit davon entfernt General Purpose zu sein. Eine nächste Innovation war dann der auch hochgradig parallel, man hat also 128 Lese-Schreibköpfe gehabt, man hat eine Transferrate von 62 MB pro Sekunde gehabt, Zugriffszeit war natürlich ein bisschen groß, 20 Millisekunden, man musste diese Köpfe erstmal positionieren und trotzdem hat man es geschafft, alle Processing Memories in 16 Millisekunden zu laden, wenn man positioniert war. wäscheschleuder die sie schon mal vielleicht bei ihrer oma gesehen haben im keller irgendwo also auch von der von der größe her ich habe so ein ding mal in live gesehen keine ahnung und meter ungefähr hoch und so 50 cm durchmesser das sind hier die schreis schreib lese köpfe und die wand dieser großen trommel das war praktisch der speicher feinmechanisch also durchaus ansprechen sein, dass diese Trommel hier nicht irgendwie geeiert hat, damit man eben in den Spuren seine Daten immer wieder gefunden hat. Gut, die Arbeitsweise der Maschine ist klar, die Control Unit schickt eben eine Instruktion an alle, macht einen Broadcast und die Möglichkeit, die die eben haben, teilnehmen oder nicht

teilnehmen. Man hat 64-Bit Floating Point Vector Operationen durchgeführt, Peak war 4 Megaflops ungefähr, 260 Megaflops für Additionen, aber im Mittel so 4 Megaflops erreicht. Ja, schwierig war für solche Systeme immer so ein Betriebssystem zu bauen, ja, es ist ein komplett neues System, muss ich erstmal bauen, und dann eben Hochsprachen neu zu entwickeln, die diese Parallelisierung auch widerspiegeln. Also man hat da so ein neues Konstrukt entwickelt, so ein For All für die automatische Parallelisierung der Laufschleife, diese Maschine auch programmieren konnte. Letztendlich, was waren die Lessons learned aus diesem Projekt? Okay, Hardware haben wir sehr viel gelernt. Wenn Sie sich anschauen, was erstmalig in diesem Projekt an Hardware-Einsatz geleistet wurde. Und wir haben auch gelernt, dass es eben für so eine komplett neue Architektur schwierig ist, Software zu entwickeln, dass das länger dauert. Also dementsprechend hat die Software Defizite. Die Compiler waren schwer zu implementieren. Also das ist alles, was so an Problemen aufgetreten ist. Ja, es gab auch keinerlei Redundanz. Das heißt, das war eine komplexe Maschine und wenn ein Bauteil ausgefallen ist, ist die ganze Maschine gestanden. Das ist natürlich bei so komplexer Hardware auch 50 Processing Units mal aufbauen, aber letztendlich ist es dann bei diesen 64 Processing Units geblieben.