

Praktikum „Rechnergestützter Entwurf digitaler Systeme“
Wintersemester 2025

Netze parallel mit dem Lee-Algorithmus in Echtzeit verdrahten

In der jungen Programmiersprache Mojo

Marvin Wollbrück
tinf105022@fh-wedel.de
Fachrichtung Technische Informatik

16. Dezember 2025



Dieses Programm wurde für die Veranstaltung "Prakt. Rechnergestützter Entwurf digitaler Systeme" der Fachhochschule Wedel entwickelt. Dabei wurde eine aus drei Aufgaben ausgewählt: Clustering, Platzierung oder Verdrahtung. Diese unterscheiden sich in ihre jeweilige Syntheseschritte und entsprechender Bewertung. Grundsätzlich wird von jeder Aufgabe verlangt, dass ein vorgestellter Algorithmus aus der gleichnamigen Vorlesung, "Rechnergestützter Entwurf digitaler Systeme", ausgewählt wird und mit einer eigenen kreativen Komponente erweitert wird.

Das fertige Programm wird mit vorgefertigten Benchmarks getestet, analysiert und bewertet. Die Ergebnisse werden mit dem VPR-Programm (*Versatile Place and Route*), von Vaughn Betz und Jonathan Rose, in der Version 4.30 verglichen. Dabei stehen ingenieurtechnisch-wissenschaftliche Kriterien im Fokus.

Das Programm implementiert den Lee-Algorithmus bzw. Maze-Algorithmus. Als kreative Komponente werden mehrere Netze in Echtzeit parallel verdrahtet. Zudem wurde dies in der noch jungen Programmiersprache Mojo von Modular implementiert.

Inhaltsverzeichnis

1 Aufgabe	1
1.1 Benchmarks	2
1.2 Algorithmus	3
1.3 Kreativer Anteil	4
2 Implementierung	6
2.1 Umgebung	6
2.2 Umsetzung	6
3 Analyse	8
3.1 Kanalbreite	8
3.2 Kritischer Pfad	10
3.3 Laufzeit	12
3.4 Fazit	13
Quellenverzeichnis	14
A Anhang	15
A.1 Benchmarksergebnisse	15
A.2 Quellcode	36

Abbildungsverzeichnis

1.1	Darstellung des Lee-Algorithmus, Pfad ermittlung von (Q)uelle zur (S)enke	3
1.2	Darstellung des parallelen Lee-Algorithmus, Pfad ermittlung von (Q)uelle zur (S)enke mit 3 (N)etze und einem (G)lobalen(R)aster .	5

Tabellenverzeichnis

1.1	Eckdaten der Benchmarks	2
3.1	Vergleich der minimalsten Kanalbreiten	8
3.2	Vergleich der Kanalbreiten	9
3.3	Vergleich der kritischen Pfade	10
3.4	Vergleich der Laufzeiten	12
A.1	Legende	15
A.2	alu4	16
A.3	apex2	17
A.4	apex4	18
A.5	bigkey	19
A.6	clma	20
A.7	des	21
A.8	diffeq	22
A.9	dsip	23
A.10	elliptic	24
A.11	ex5p	25
A.12	ex1010	26
A.13	frisc	27
A.14	misex3	28
A.15	pdc	29
A.16	s298	30
A.17	s38417	31
A.18	s38584.1	32
A.19	seq	33
A.20	spla	34
A.21	tseng	35

Abkürzungsverzeichnis

VPR	<i>versatile place and route</i>
BLE	<i>basic logic element</i>
LUT	<i>lookup table</i>

1 Aufgabe

Aus der Vorlesung ?Rechnergestützter Entwurf digitaler Systeme? oder Literatur soll ein Algorithmus aus den Syntheseschritte Clustering, Platzierung oder Verdrahtung der Layoutsynthese gewählt werden. Der gewählte Algorithmus darf nicht 1:1 Umgesetzt werden, stattdessen muss der Algorithmus einen kreativen Anteil beinhalten. Zusätzlich müssen Ein- und Ausgabedateien in einem vorgegebenen Format umgesetzt werden. Hierbei wird das Format vom Vergleichsprogramm *versatile place and route* (VPR), von Vaughn Betz und Jonathan Rose, der Version 4.30 vorgegeben.

Die Entscheidung fiel auf den Syntheseschritt Verdrahtung. In dieser sollen 20 vorgegebenen Benchmarks mit dem kreativ erweiterten Algorithmus auf minimale Kanalbreite verdrahtet werden. Die damit vorgegebenen Netz- und Platzierungslisten beinhalten *basic logic element* (BLE) mit maximal einer *lookup table* (LUT) und einem Flipflop.

Anschließend, werden folgende Kriterien mit dem Vergleichsprogramm VPR verglichen:

1. Laufzeit des Algorithmus
2. Kritischer Pfad
3. Minimale Breite der Verdrahtungskanäle

1.1 Benchmarks

Tabelle 1.1: Eckdaten der Benchmarks

Benchmarks	Rastergröße	Blöcke	Netze	davon Globalenetze
alu4	40x40	1544	1536	0
apex2	44x44	1919	1916	0
apex4	36x36	1290	1271	0
bigkey	54x54	2133	1936	1
clma	92x92	8527	8445	1
des	63x63	2092	1847	0
diffeq	39x39	1600	1561	1
dsip	54x54	1796	1599	1
elliptic	61x61	3849	3735	1
ex5p	33x33	1135	1072	0
ex1010	68x68	4618	4608	0
frisc	60x60	3692	3576	1
misex3	38x38	1425	1411	0
pdc	68x68	4631	4591	0
s298	44x44	1941	1935	1
s38417	81x81	6541	6435	1
s38584.1	81x81	6789	6485	1
seq	42x42	1826	1791	0
spla	61x61	3752	3706	0
tseng	33x33	1221	1099	1

1.2 Algorithmus

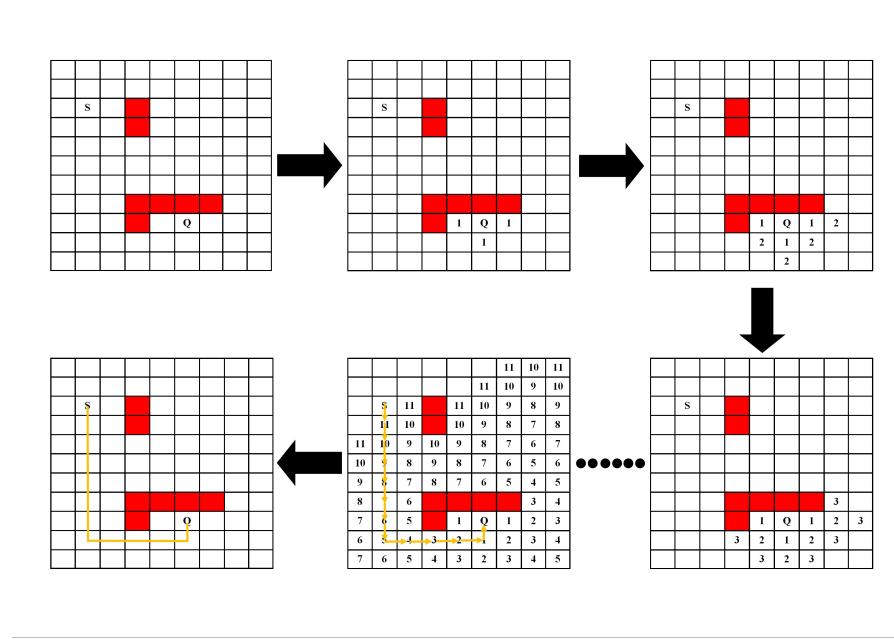


Abbildung 1.1: Darstellung des Lee-Algorithmus, Pfad ermittlung von (Q)uelle zur (S)enke

Der gewählte Algorithmus, welcher für die Aufgabe genutzt wird, ist der Lee-Algorithmus (*maze routing*). Dieser sucht über die gesamte Fläche die kürzeste Verbindung zweier Punkte in einem Raster. Dabei wird der Algorithmus in drei Phasen aufgeteilt, Initiierung exkludiert.

Das Raster ist beim Start der Initiierung leer. Nun werden die Quelle und die Senken eines Netzes im Raster markiert. Zudem werden bereits bestehende Verbindungen desselben Netzes auch als Quelle markiert, während Verbindungen anderer Netze als unpassierbar markiert sind.

Die Ausbreitungsphase ist die erste Phase. In dieser breitet sich eine Wellenfront, beginnend bei der Quellen Markierung, sukzessiv auf benachbarte Rasterpunkte aus. Dabei werden benachbarte Punkte von bereits indiziertem Punkten aufsteigend indiziert. Dabei stellt die Quelle den Startindex dar. Die Wellenfront schreitet solange voran bis eine Senke erreicht ist.

Beim erreichen einer Senke beginnt die Rückverfolgungsphase. Hierbei wird die Wellenfront von der Senke zur Quelle in Richtung absteigender Indizes rückver-

folgt. Dabei wird die kürzeste Verbindung berücksichtigt, sollte es mehrere kürzeste Verbindungen geben, wird die mit den wenigsten Richtungswechseln bevorzugt.

Zum Schluss werden in der Aufräumphase die Rasterpunkte der gefundenen Verbindung als belegt markiert, die restlichen Indizierungen werden gelöscht.

1.3 Kreativer Anteil

Der kreative Anteil sieht die Echtzeit-Parallelisierung des Algorithmus vor. Dabei sollen die Netze parallel verdrahtet werden. Hierbei werden gemeinsame Ressourcen genutzt, diese müssen abgesicherte Zugriffe haben, damit keine Wettlaufsituationen entstehen. Jeder Schreibzugriff wird die anderen Threads blockieren und somit auch die Verdrahtung dieser pausieren. Zudem wird die Verdrahtungsreihenfolge und -geschwindigkeit zusätzlich vom Scheduler mit beeinflusst. Hierbei wird der Scheduler zu einem Zufallsgenerator, da dieser auch alle anderen Prozesse auf der ausführenden Maschine mitplanen muss, unter der Annahme, dass mehr Threads als Rechenkerne vorhanden sind.

Sollte nun der Scheduler ideal funktionieren, dass dieser eine perfekte Gleichverteilung von Rechenzeit den Verdrahtungsthreads zuweist, sodass jeder parallel eine atomare Aktion ausführt, können Annahmen gemacht werden. Dabei wird beachtet, dass eine Verdrahtung einer Senke eines Netzes andere Netze blockieren kann.

1. Sollten für alle Netze die aktuelle Suche nach einer Verdrahtung zu einer Senke neugestartet werden, sobald ein Netz eine Senke verdrahtet hat. So wird immer die aktuell kürzeste mögliche Verdrahtung, entsprechend der aktuellen Blockierungen von bestehenden Verdrahtungen, verdrahtet. Womit es einer Vorsortierung der Netze von kürzesten zu längsten, aber mit ständigeren neu sortieren, gleichkommt.
2. Findet kein Neustart der Verdrahtungssuche statt, wie oben, so muss bei der Rückverfolgung mögliche Blockierungen berücksichtigt werden. Diese führen, wenn unvermeidbar, zu einem Neustart der Suche. Hierbei werden die Verdrahtungen mit der aktuellen Restverdrahtungslänge, die am kürzesten ist, immer zuerst auf mögliche Verdrahtung geprüft und eventuell verdrahtet, wenn nicht blockiert. Diese Vorgehensweise verdrahtet damit längere Verdrahtungen vor den aktuellen kürzesten Verdrahtungen und erzeugt eine Durchmischung der Verdrahtungen, anstatt der Sortierung nach den immer aktuell kürzesten.

Es wird für die Implementierung die zweite Variante gewählt. Durch die Durchmischung der Verdrahtungen wird angenommen, dass Netze mit langen oder vielen

Verdrahtungen sich nicht überwiegend blockieren, was zu einer besseren Nutzung der Kanalbreite führen könnte.

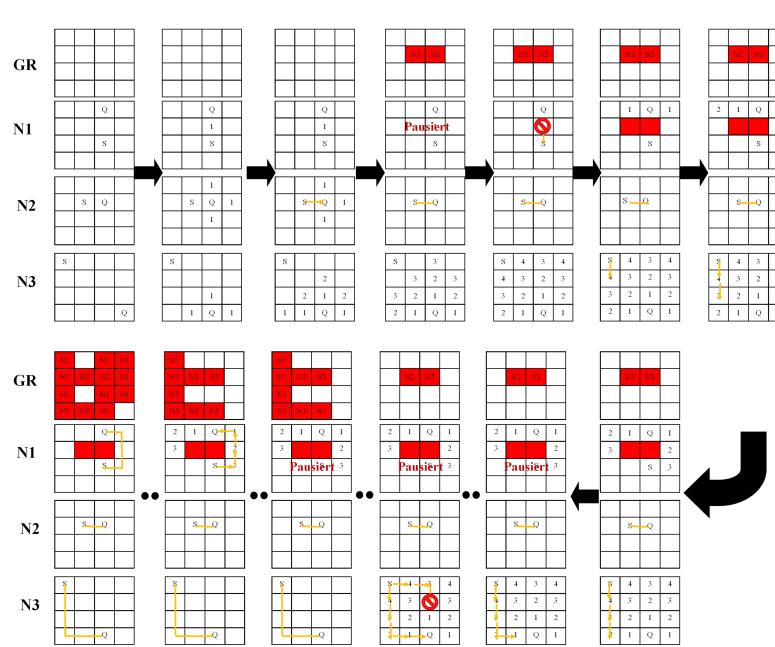


Abbildung 1.2: Darstellung des parallelen Lee-Algorithmus, Pfad ermittlung von (Q)uelle zur (S)enke mit 3 (N)etze und einem (G)lobalen(R)aster

2 Implementierung

Zur Implementierung wurde die junge Programmiersprache Mojo von Modular gewählt. Diese wird explizit für hoch performanter KI-Infrastruktur und Heterogene Hardware entwickelt. Dabei wird nicht durch den Compiler Annahmen getätigt welche Hardware genutzt werden soll, stattdessen werden hierfür explizite Bibliotheken entwickelt.

Die Bibliotheken für die Nutzung einer GPU setzt Hardware voraus die CUDA unterstützen. Dadurch würde das Programm nur auf bestimmter Hardware ausgeführt werden können. Deswegen werden CPU-Bibliotheken für die Echtzeit Parallelisierung genutzt.

2.1 Umgebung

Es wurde auf einem Windows 11 Computer mit einer CPU mit 4 physischen Kernen, 11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz, und mit 32GB Arbeitsspeicher entwickelt und getestet. Wobei eine Unix Umgebung genutzt werden muss für Mojo, weswegen Ubuntu als Windows Subsystem für Linux mit effektivem Arbeitsspeicher von 16GB genutzt wurde.

Die Entwicklung startete bei der Mojo Version v24.6 und bis v25.5 sukzessiv aktualisiert wurde.

2.2 Umsetzung

Für die Umsetzung von Echtzeit Parallelisierung können zwei Bibliotheken genutzt werden. Aus der Bibliothek *algorithm/functional* ist die Funktion *parallelize* für die parallele Ausführung einer Funktion in sub-Tasks implementiert. Diese blockiert den Hauptthread und wartet bis n-Workers einer Funktion abgeschlossen sind. Zudem lässt sich diese Funktion in ein Struct, Klassen ähnliches Objekt, einpflegen.

In der Bibliothek *runtime/asyncrt* befinden sich das Struct Task, welches mit einer *coroutine*, eine spezielle Funktion, erstellt wird und diese dann asynchron Ausführt. Dies lässt sich nicht in einem Struct implementieren.

Während der Implementierung stellte sich heraus, dass beide Bibliotheken kein Hyperthreading unterstützen und nur maximal 8 Threads nutzen. Zudem blockiert ein Task, bei beiden Bibliotheken, immer ein Thread bis dieser freigegeben wird. Damit werden die ersten 8 Task aus einer Reihenfolge zuerst abgearbeitet, aber da-

nach folgen per Zufall durch den Scheduler weitere Task aus derselben Reihenfolge bis alle abgearbeitet sind.

Aufgrund dessen, dass beide gleich parallelisiert, aber *parallelize* direkt beim Aufruf wartet und in einen Struct eingepflegt werden kann, wird dessen Bibliothek genutzt.

Eine Kanalbreite besitzt mehrere Leitungsebenen, sofern nicht die Breite 1 ist. Diese parallel für ein Netz zu nutzen würde zu einem erhöhten Speicherbedarf führen in Kombination mit den Netzen die sowieso parallel Verdrahtet werden. Daher werden die Leitungsebenen sukzessiv durchgegangen, wenn die aktuelle keine Verdrahtung mehr ermöglicht.

Die Rückverfolgung der möglichen Pfade werden zur Laufzeitkürzung nicht jeden möglichen Pfad zurückverfolgen. Stattdessen Wird ein Pfad nach dem ersten abknicken gerade zurückverfolgt bis ein abknicken notwendig wird. Hierbei wird angenommen, dass ein Pfad mit den wenigsten Knicken seine Knicke an den äußersten Bereichen hat.

3 Analyse

Die Benchmarks wurden jeweils 20mal mit VPR und dem parallelen Lee-Algorithmus verdrahtet. Zusätzlich wurden die Benchmarks 5mal mit dem Lee-Algorithmus sequentiell verdrahtet, dies dient nur als Referenz für Anschaulichkeitszwecke.

Die Ergebnisse der Verdrahtungen sind im Anhang im Detail aufgeführt. Für den Vergleich zwischen dem erweiterten Algorithmus und VPR werden die Mittelwerte genutzt.

Die Erkenntnis aus der Implementierung, dass nur 8 Netze in Echtzeit parallel verdrahtet werden, wird berücksichtigt. Zudem wird nicht vorsortiert, wodurch die ersten 8 Netze immer gleich mit den zuerst auftretenden nicht globalen Netzen in einer Netzliste sind.

3.1 Kanalbreite

Tabelle 3.1: Vergleich der minimalsten Kanalbreiten

Benchmark	chan_width			Faktor		
	vpr430	parallel lee	sequentiell lee	parallel/vpr	sequentiell/parallel	sequentiell/vpr
alu4	11	16	16	1,454545455	1	1,454545455
apex2	12	18	19	1,5	1,055555556	1,583333333
apex4	13	20	22	1,538461538	1,1	1,692307692
bigkey	6	11	13	1,833333333	1,181818182	2,166666667
clma	13	20	22	1,538461538	1,1	1,692307692
des	8	13	15	1,625	1,153846154	1,875
diffeq	8	13	14	1,625	1,076923077	1,75
dsip	6	10	11	1,666666667	1,1	1,833333333
elliptic	11	18	20	1,636363636	1,111111111	1,818181818
ex5p	14	20	21	1,428571429	1,05	1,5
ex1010	11	17	18	1,545454545	1,058823529	1,636363636
frisc	13	19	20	1,461538462	1,052631579	1,538461538
misex3	12	18	18	1,5	1	1,5
pdc	18	26	26	1,444444444	1	1,444444444
s298	8	13	14	1,625	1,076923077	1,75
s38417	8	14	15	1,75	1,071428571	1,875
s38584.1	9	14	15	1,555555556	1,071428571	1,666666667
seq	12	18	19	1,5	1,055555556	1,583333333
spla	14	22	23	1,571428571	1,045454545	1,642857143
tseng	7	12	14	1,714285714	1,166666667	2
				1,575705544	1,076408309	1,700140138

Tabelle 3.2: Vergleich der Kanalbreiten

Benchmark	chan_width			Faktor		
	vpr430	parallel lee	sequentiell lee	parallel/vpr	sequentiell/parallel	sequentiell/vpr
alu4	11	16	16	1,454545455	1	1,454545455
apex2	12	18,85	19	1,570833333	1,00795756	1,583333333
apex4	13	20,35	22	1,565384615	1,081081081	1,692307692
bigkey	6	11,2	13	1,866666667	1,160714286	2,166666667
clma	13	20	22	1,538461538	1,1	1,692307692
des	8	13,9	15	1,7375	1,079136691	1,875
diffeq	8	13,8	14	1,725	1,014492754	1,75
dsip	6	10	11	1,666666667	1,1	1,833333333
elliptic	11	18	20	1,636363636	1,111111111	1,818181818
ex5p	14	20,05	21	1,432142857	1,047381546	1,5
ex1010	11	17,6	18	1,6	1,022727273	1,636363636
frisc	13	19,85	20	1,526923077	1,007556675	1,538461538
misex3	12	18	18	1,5	1	1,5
pdc	18	26	26	1,444444444	1	1,444444444
s298	8	13,15	14	1,64375	1,064638783	1,75
s38417	8	14,4	15	1,8	1,041666667	1,875
s38584.1	9	14	15	1,555555556	1,071428571	1,666666667
sed	12	18,05	19	1,504166667	1,052631579	1,583333333
spla	14	22,05	23	1,575	1,0430839	1,642857143
tseng	7	12	14	1,714285714	1,166666667	2
				1,602884511	1,058613757	1,700140138

Aus den Tabellen ist erkennbar, dass der Algorithmus im vergleich zu VPR eine ~1,60-fache oder eine ~1,58-fache, bei minimalstem Ergebnis, größere minimale Kanalbreite erreicht. Zudem erzielt die sequentielle Ausführung zur parallelen Ausführung des Algorithmus eine ~1,06-fache größere Kanalbreite, bzw. parallel erreicht das ~0,95-fache von sequentiell.

Damit spart die Parallelisierung potentiell ~5% Leitungen pro Kanal zur sequentiellen Ausführung. Dennoch benötigt diese ~60% mehr Leitungen pro Kanal als VPR.

VPR nutzt den *Pathfinder negotiated congestion algorithm*. Dieser ermittelt zuerst die kürzesten Verdrahtungen ungeachtet der Kanalbreite oder Logikblockpins. Danach wird iterativ anhand einer Kostenfunktion alternative Verdrahtungen gesucht, bis keine Überbeanspruchung von Kanalbreite und Pins vorliegt. Kosten ergeben sich durch die aktuelle Überbeanspruchung und der Vorherigen. Dadurch werden die Kosten eines Kanals stetig gesteigert um alternative Verdrahtungen zu erzwingen.

Während der Lee-Algorithmus die kürzeste Verdrahtung ermittelt, anhand von bestehenden Verdrahtungen anderer Netze und die gefundene Verdrahtung unveränderlich verdrahtet, womit andere Netze blockiert werden und so eine größere

Kanalbreite benötigt wird.

Des Weiteren gibt es Kanalbreiten beim parallelen Algorithmus Kommastellen, welche eigentlich nicht geben dürfte, da eine Kanalbreite eine Ganzzahl sein muss. Dies ist das Ergebnis von unterschiedlichen minimalste Kanalbreiten der 20 Ausführungen. Damit zeigt es, dass der Scheduler als Zufallskomponente gewertet wird, denn es wird keine Sortierung vorgenommen, sondern die Reihenfolge aus den Netzlisten übernommen. Nach den ersten 8 Netzen wird durch den Scheduler zufällig ein nächstes Netz zum Verdrahten gestartet.

3.2 Kritischer Pfad

Tabelle 3.3: Vergleich der kritischen Pfade

Benchmark	crit_t(s)			Faktor		
	vpr430	parallel lee	sequentiell lee	parallel/vpr	sequentiell/parallel	sequentiell/vpr
alu4	1,12E-07	1,73E-07	1,58E-07	1,546351935	0,914808286	1,414615564
apex2	1,41E-07	1,50E-07	1,56E-07	1,06506711	1,039704545	1,107355116
apex4	1,31E-07	1,49E-07	1,33E-07	1,131570888	0,892237485	1,009629963
bigkey	1,06E-07	7,75E-08	5,84E-08	0,732117464	0,753712199	0,551805864
clma	2,54E-07	1,72E-07	1,69E-07	0,678731601	0,981480685	0,666161957
des	1,21E-07	2,16E-07	1,17E-07	1,775718979	0,541591804	0,961714846
diffeq	8,82E-08	1,13E-08	1,10E-08	0,127861646	0,971700121	0,124243177
dsip	7,98E-08	5,72E-08	5,07E-08	0,717556123	0,88565328	0,635505934
elliptic	1,83E-07	1,48E-08	1,64E-08	0,081103522	1,109685436	0,089999398
ex5p	1,09E-07	1,29E-07	1,31E-07	1,186312596	1,011594951	1,200067832
ex1010	2,48E-07	2,95E-07	2,66E-07	1,192284323	0,900935547	1,074171329
frisc	1,69E-07	6,41E-08	6,16E-08	0,379409674	0,962282515	0,365099296
misex3	1,06E-07	1,29E-07	1,33E-07	1,210495476	1,036756919	1,25498956
pdc	2,39E-07	2,83E-07	2,90E-07	1,186391642	1,022529382	1,213120312
s298	1,98E-07	1,20E-07	1,14E-07	0,605142756	0,948371796	0,573900323
s38417	1,77E-07	1,22E-07	1,03E-07	0,687317431	0,845670403	0,581244008
s38584.1	1,27E-07	1,80E-07	1,89E-07	1,419643223	1,04972728	1,490238219
seq	1,27E-07	1,58E-07	1,73E-07	1,241811751	1,09815134	1,363697238
spla	1,76E-07	2,45E-07	2,60E-07	1,389363164	1,061831457	1,475269513
tseng	9,00E-08	3,23E-08	3,76E-08	0,358544189	1,165964564	0,418049819
				0,935639775	0,9597195	0,878543963

Durchschnittlich wird ein kritischer Pfad erreicht, der das $\sim 0,94$ -fache von VPR ist. Allerdings wird ersichtlich, dass Benchmarks mit einem globalen Netz, also Vollsynchrong, überwiegend einen niedrigeren kritischen Pfad als VPR haben. Während die Benchmarks ohne globale Netze überwiegend einen höheren kritischen Pfad haben.

Dies ist ein weiteres Resultat der Unterschiede der Verdrahtungsalgorithmen. Hierbei werden, anders als beim VPR, Verdrahtungen auf derselben Leitungsebe-

ne verlängert die später verdrahtet wurden. Eine Verdrahtung wird erst auf die nächste Leitungsebene verlegt, wenn für das Netz keine Verdrahtungen auf der aktuellen Ebene möglich sind. Zudem wird immer auf einer Verdrahtungsebene der mögliche kürzeste Pfad gewählt.

Während bei VPR durch die Überbeanspruchung in der Kostenfunktion bereits die Leitungen pro Kanal berücksichtigt werden. Dadurch wird ein niedriger kritischer Pfad bei niedriger Kanalbreite erzielt.

In einem vollsynchrone Entwurf sind alle Speicherglieder mit demselben Takt gesteuert. Dadurch muss die Änderung eines Eingangs nicht innerhalb eines Taktes einen Ausgang ändern. Die Änderung durchläuft Teilabschnitte, getrennt durch die Speicherglieder, eines Pfades von Eingang zum Ausgang. Diese Teilabschnitte bilden eigene kritische Pfade und der Längste bestimmt den endgültigen kritischen Pfad.

Die parallele Verdrahtung begünstigt den kritischen Pfad eines vollsynchrone Entwurfs, da die Unterteilungen innerhalb, entsprechend dem Algorithmus und einer höheren Kanalbreite, im Durchschnitt kürzere Verdrahtungen aufweisen. Hierbei gibt es eine Abhängigkeit durch die Platzierung, welche die Logikblöcke entsprechend den Netzen günstig in Nähe zueinander platziert.

Die sequentielle Ausführung hat die durchschnittlichen niedrigsten kritischen Pfade, während es die höchsten Kanalbreiten hat. Dies zeigt die grundlegende Auswirkung der Kanalbreite auf den kritischen Pfad.

3.3 Laufzeit

Tabelle 3.4: Vergleich der Laufzeiten

Benchmark	Laufzeit(s)			Faktor		
	vpr430	parallel lee	sequentiell lee	parallel/vpr	sequentiell/parallel	sequentiell/vpr
alu4	46	92,732	147,2122	2,01319523	1,587501618	3,195950684
apex2	72,93225	125,6391	201,1956	1,722682352	1,601377278	2,758664377
apex4	10,2227	97,0542	213,124	9,493988868	2,195927636	20,84811253
bigkey	200,88325	186,3637	172,4268	0,92772145	0,92521666	0,858343341
clma	106,09355	2237,3856	6773,7968	21,08879946	3,027550012	63,84739506
des	21	176,30165	519,8888	8,451033121	2,948859526	24,92090952
diffeq	9,66465	81,90045	133,5402	8,474228244	1,630518514	13,81738604
dsip	161,60915	128,858	281,222	0,797342848	2,182419549	1,74013662
elliptic	142,8438	774,63705	1216,5156	5,422965855	1,570433018	8,516404632
ex5p	11,19885	87,26125	134,0234	7,791983105	1,535886777	11,96760382
ex1010	262,5877	841,3176	1639,3188	3,203949004	1,948513617	6,242938264
frisc	26,14425	484,41445	1065,3874	18,52852731	2,199330346	40,75035237
misex3	52,4103	130,01025	213,1402	2,480624038	1,639410739	4,066761686
pdc	351,9734	1246,66775	2670,7008	3,541937402	2,142271507	7,587791577
s298	27,97755	95,02495	182,351	3,396471457	1,918980226	6,517761562
s38417	52,49685	1152,5487	2475,1094	21,95462585	2,147509602	47,14776982
s38584.1	99,26635	1116,244	3019,644	11,24493849	2,705182738	30,41961349
seq	63,9462	184,1316	198,5752	2,87947681	1,078441723	3,105347933
spla	76,5445	635,4968	866,7878	8,302318259	1,363953052	11,32397233
tseng	2,5784	73,7326	83,7522	28,59626125	1,135891044	32,48223705
				8,51565352	1,874258759	17,10577264

Die Laufzeit beginnt bei Start der Ausführung eines Programms bis zu derer Beendigung.

Beide Programme nutzen maximal 30 Fehlversuche um ein Benchmark für eine Kanalbreite erfolgreich zu Verdrahten. Dies wiederholt sich bis eine binäre Suche die kleinste mögliche Kanalbreite gefunden hat, beginnend bei einer Kanalbreite von 12.

Der Algorithmus braucht $\sim 8,51$ -mal so lang wie VPR. Womit die Echtzeit Parallelisierung kein zeitlicher Gewinn darstellt, im Vergleich zu VPR.

Allerdings benötigt der parallele Algorithmus ungefähr nur die Hälfte der Zeit als der Sequentielle. Zusätzlich ist anzumerken, dass die Sequentielle Ausführung immer dieselbe Reihenfolge der Netze nutzt, womit keine 30 Fehlversuche pro Kanalbreite braucht, da immer dasselbe Ergebnis ermittelt wird und somit 29-mal sich Wiederholt. Dies war nur notwendig für die Vergleichbarkeit.

Die parallele Ausführung wiederrum ermittelt andere Ergebnisse bei selber Kanalbreite und könnte mit Erhöhung der Fehlversuche eine erfolgreiche Verdrahtung generieren. Dies hängt somit vom Scheduler ab, welche Netze häufiger bzw.

früher Rechenzeit erhält. Damit gibt es eine Zufallskomponente.

3.4 Fazit

In Summe schneidet der parallelisierte Lee-Algorithmus schlechter als der *Pathfinder negotiated congestion algorithm* von VPR ab, da der parallele Lee-Algorithmus annähernd einen gleichen kritischen Pfad bei größerer Kanalbreite nach längerer Laufzeit ermittelt. Denn es gibt keine Bewertung von Verdrahtungen, die zu einer Alternierung der Verdrahtungen und somit zu Kompromissen führen kann. Sondern es wird vom Scheduler abhängig gemacht, welches sich als eine Zufallskomponente verhält.

Zudem werden bedingt durch die Programmiersprache Mojo nicht alle Netze in Echtzeit parallel Verdrahtet, sondern nur 8 Netze zurzeit. Sobald ein Netz fertig Verdrahtet ist wird die nächste Verdrahtung eines Netzes durch den Scheduler zufällig gestartet.

Die Limitierung auf 8 Verdrahtungsthreads geht auf die Optimierung für die Laufzeit eines Mojo-Programms, seitens der Mojo-Entwickler, zurück. Bei Wahl einer anderen Programmiersprache, die Hyperthreading erlaubt, würde die Annahme vom kreativen Anteil dazu führen, dass die Laufzeit sich erhöht, weil häufiger Blockierungen für andere Netze entstehen und so die Suche entsprechend häufiger neugestartet werden müsste. Dennoch könnte sich der Kritische Pfad eventuell verbessern, wenn alle Netze von Anfang an beim Wettlauf um die kürzeste Verdrahtung dabei sind.

Die Verdrahtungssuchen durchlaufen die Leitungsebenen einer Kanalbreite, wenn auf der aktuellen keine weitere Verdrahtung möglich ist. Dadurch werden Verdrahtungen gezwungenermaßen verlängert, obwohl eine mögliche kürzere Möglichkeit auf einer anderen Leitungsebene besteht. Die Berücksichtigung der anderen Leitungsebenen von Beginn an, könnte zu einer Verkleinerung der Kanalbreite beitragen.

Diese möglichen Erweiterungen beinhalten weiterhin den Zufallsfaktor durch den Scheduler, somit könnte durch Zufall eine niedrigere Kanalbreite erreicht werden als bei VPR. Dennoch zeigt VPR mit dessen Algorithmus, welcher anhand einer Kostenfunktion, zuverlässiger ist für eine niedrige Kanalbreite in annehmbarer Laufzeit.

Quellenverzeichnis

- [1] BETZ, Vaughn und ROSE, Jonathan: *VPR: A New Packing, Placement and Routing Tool for FPGA Research*. University of Toronto, Toronto, 1997.
- [2] SAWITZKI, Sergei: *Rechnergestützter Entwurf digitaler Systeme [Vorlesungs-Handout]* FH Wedel, Wedel, 2024.

A Anhang

A.1 Benchmarksergebnisse

Tabelle A.1: Legende

Mittelwert	MAX	MIN
------------	-----	-----

Tabelle A.2: alu4

#	vpr430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	11	1,12E-07	0	46	16	1,75E-07	1	33,29	16 1,58E-07
2	11	1,12E-07	0	46	16	1,64E-07	1	23,151	16 1,58E-07
3	11	1,12E-07	0	46	16	1,56E-07	1	32,594	16 1,58E-07
4	11	1,12E-07	0	46	16	1,72E-07	1	56,274	16 1,58E-07
5	11	1,12E-07	0	46	16	2,27E-07	2	22,708	16 1,58E-07
6	11	1,12E-07	0	47	16	1,94E-07	1	33,431	
7	11	1,12E-07	0	46	16	1,43E-07	1	40,937	
8	11	1,12E-07	0	46	16	1,96E-07	1	44,789	
9	11	1,12E-07	0	46	16	1,54E-07	1	24,938	
10	11	1,12E-07	0	47	16	1,47E-07	1	27,272	
11	11	1,12E-07	0	46	16	1,96E-07	1	30,896	
12	11	1,12E-07	0	46	16	1,75E-07	1	23,224	
13	11	1,12E-07	0	47	16	1,93E-07	1	25,652	
14	11	1,12E-07	0	46	16	1,56E-07	1	25,635	
15	11	1,12E-07	0	46	16	1,45E-07	1	23,188	
16	11	1,12E-07	0	46	16	1,87E-07	1	26,726	
17	11	1,12E-07	0	45	16	1,53E-07	1	24,327	
18	11	1,12E-07	0	46	16	1,83E-07	1	23,167	
19	11	1,12E-07	0	45	16	1,67E-07	1	29,066	
20	11	1,12E-07	0	46	16	1,68E-07	1	23,375	
	11	1,12E-07	0	46	16	1,73E-07	1,05	29,732	16 1,58E-07
	x20	1,12E-07	Gesamt:	46	16	2,27E-07	Gesamt:	92,732	16 1,58E-07
	x20	1,12E-07		x20	1,43E-07		x5	1,58E-07	Gesamt: 147,2122

Tabelle A.3: apex2

#	vpr430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	12	1,41E-07	1	13,661	18	1,38E-07	3	51,045	19 1,56E-07
2	12	1,41E-07	1	14,199	19	1,34E-07	1	54,043	19 1,56E-07
3	12	1,41E-07	1	13,129	19	1,49E-07	1	50,094	19 1,56E-07
4	12	1,41E-07	1	12,578	19	1,43E-07	1	50,644	19 1,56E-07
5	12	1,41E-07	1	12,263	19	1,58E-07	1	45,694	19 1,56E-07
6	12	1,41E-07	1	13,05	19	1,43E-07	1	48,624	
7	12	1,41E-07	1	12,738	19	1,63E-07	1	50,538	
8	12	1,41E-07	1	13,049	19	1,59E-07	1	46,264	
9	12	1,41E-07	1	13,363	19	1,61E-07	1	48,586	
10	12	1,41E-07	1	13,198	19	1,46E-07	1	47,395	
11	12	1,41E-07	1	11,88	19	1,48E-07	1	50,308	
12	12	1,41E-07	1	12,81	19	1,61E-07	1	48,116	
13	12	1,41E-07	1	13,18	19	1,43E-07	1	48,933	
14	12	1,41E-07	1	12,842	18	1,54E-07	3	14,003	
15	12	1,41E-07	1	12,373	19	1,35E-07	1	50,563	
16	12	1,41E-07	1	13,321	19	1,43E-07	1	47,779	
17	12	1,41E-07	1	12,733	19	1,73E-07	1	48,435	
18	12	1,41E-07	1	12,399	18	1,43E-07	3	53,765	
19	12	1,41E-07	1	12,995	19	1,48E-07	1	49,77	
20	12	1,41E-07	1	12,884	19	1,54E-07	1	48,183	
	12	1,41E-07	1	12,93225	18,85	1,50E-07	1,3	47,6391	19 1,56E-07
x20	1,41E-07								3 21,1956
	12	1,41E-07	Gesamt:	72,93225	18	1,73E-07	Gesamt:	125,6391	19 1,56E-07
	x20	1,41E-07			x3	1,34E-07		x5	1,56E-07 Gesamt: 201,1956

Tabelle A.4: apex4

#	vpr430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	13	1,31E-07	0	10,271	20	1,66E-07	1	59,132	22
2	13	1,31E-07	0	10,172	20	1,55E-07	1	28,153	22
3	13	1,31E-07	0	10,169	20	1,59E-07	1	49,076	22
4	13	1,31E-07	0	10,112	20	1,48E-07	1	40,968	22
5	13	1,31E-07	0	10,588	21	1,38E-07	1	30,1	22
6	13	1,31E-07	0	10,34	21	1,35E-07	1	31,425	
7	13	1,31E-07	0	10,198	20	1,42E-07	1	32,019	
8	13	1,31E-07	0	10,148	20	1,58E-07	1	34,785	
9	13	1,31E-07	0	10,226	20	1,44E-07	1	29,511	
10	13	1,31E-07	0	10,17	21	1,68E-07	1	34,436	
11	13	1,31E-07	0	10,131	21	1,50E-07	1	29,897	
12	13	1,31E-07	0	10,205	20	1,44E-07	1	55,89	
13	13	1,31E-07	0	10,105	20	1,40E-07	1	31,84	
14	13	1,31E-07	0	10,244	21	1,40E-07	1	31,281	
15	13	1,31E-07	0	10,161	20	1,57E-07	1	57,773	
16	13	1,31E-07	0	10,154	20	1,48E-07	1	41,075	
17	13	1,31E-07	0	10,229	21	1,63E-07	1	29,9	
18	13	1,31E-07	0	10,157	21	1,40E-07	1	31,848	
19	13	1,31E-07	0	10,537	20	1,35E-07	1	36,14	
20	13	1,31E-07	0	10,137	20	1,38E-07	1	25,835	
	13	1,31E-07	0	10,2227	20,35	1,49E-07	1	37,0542	22
	13	1,31E-07	0	10,2227				1,33E-07	3
x20	1,31E-07								33,124
	13	1,31E-07	Gesamt:	10,2227	20	1,68E-07	Gesamt:	97,0542	22
	x20	1,31E-07		x13	1,35E-07		x5	1,33E-07	Gesamt: 213,124

Tabelle A.5: bigkey

#	vpr430			parallel lee			sequentiell lee					
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)			
1	6	1,06E-07	3	21,414	12	9,52E-08	3	50,053	13	5,84E-08	2	50,526
2	6	1,06E-07	3	21,282	12	9,54E-08	3	46,86	13	5,84E-08	2	51,932
3	6	1,06E-07	3	21,393	11	8,60E-08	2	46,148	13	5,84E-08	2	59,405
4	6	1,06E-07	3	22,136	11	9,49E-08	3	7,725	13	5,84E-08	2	50,169
5	6	1,06E-07	3	20,927	11	9,54E-08	2	49,463	13	5,84E-08	2	50,102
6	6	1,06E-07	3	19,838	11	9,54E-08	3	13,385				
7	6	1,06E-07	3	21,104	11	5,39E-08	2	52,526				
8	6	1,06E-07	3	20,054	11	5,39E-08	2	48,584				
9	6	1,06E-07	3	20,128	11	9,54E-08	2	45,579				
10	6	1,06E-07	3	21,383	11	9,04E-08	3	0,006				
11	6	1,06E-07	3	19,882	11	9,54E-08	3	16,397				
12	6	1,06E-07	3	19,948	12	5,89E-08	3	10,609				
13	6	1,06E-07	3	20,835	11	5,39E-08	2	45,425				
14	6	1,06E-07	3	23,286	12	5,39E-08	3	53,404				
15	6	1,06E-07	3	21,16	11	9,52E-08	3	22,276				
16	6	1,06E-07	3	22,38	11	9,54E-08	3	18,553				
17	6	1,06E-07	3	21,363	11	5,39E-08	2	47,738				
18	6	1,06E-07	3	21,171	11	9,49E-08	2	58,447				
19	6	1,06E-07	3	19,123	11	5,21E-08	2	54,62				
20	6	1,06E-07	3	18,858	11	4,16E-08	2	39,476				
	6	1,06E-07	3	20,88325	11,2	7,75E-08	2,5	36,3637	13	5,84E-08	2	52,4268
x20	6	1,06E-07	200,88325	11	9,54E-08	186,3637	13	5,84E-08				172,4268
x16	6	1,06E-07	x16	4,16E-08	x5	5,84E-08						

Tabelle A.6: clma

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	13	2,54E-07	1	46,659	20	1,92E-07	36	57,045	22	1,69E-07	112	20,84
2	13	2,54E-07	1	46,53	20	1,25E-07	38	33,881	22	1,69E-07	113	27,152
3	13	2,54E-07	1	46,063	20	1,55E-07	36	14,673	22	1,69E-07	112	44,395
4	13	2,54E-07	1	46,343	20	1,64E-07	39	33,377	22	1,69E-07	112	43,016
5	13	2,54E-07	1	46,125	20	1,33E-07	36	33,162	22	1,69E-07	113	13,581
6	13	2,54E-07	1	46,687	20	2,12E-07	38	23,327				
7	13	2,54E-07	1	45,167	20	1,97E-07	36	16,285				
8	13	2,54E-07	1	45,41	20	9,21E-08	36	20,895				
9	13	2,54E-07	1	45,614	20	1,56E-07	37	19,443				
10	13	2,54E-07	1	45,749	20	1,52E-07	38	40,438				
11	13	2,54E-07	1	45,702	20	1,18E-07	36	13,251				
12	13	2,54E-07	1	46,761	20	2,80E-07	36	42,31				
13	13	2,54E-07	1	46,249	20	2,90E-07	36	30,216				
14	13	2,54E-07	1	46,132	20	1,82E-07	36	56,82				
15	13	2,54E-07	1	46,412	20	1,66E-07	36	33,385				
16	13	2,54E-07	1	46,395	20	1,93E-07	38	28,214				
17	13	2,54E-07	1	45,472	20	1,56E-07	38	20,705				
18	13	2,54E-07	1	46,182	20	1,48E-07	37	48,819				
19	13	2,54E-07	1	46,064	20	1,69E-07	36	27,039				
20	13	2,54E-07	1	46,155	20	1,62E-07	36	54,427				
	13	2,54E-07	1	46,09355	20	1,72E-07	36,75	32,3856	22	1,69E-07	112,4	29,7968
	x20	2,54E-07		106,09355	20	2,90E-07		2237,3856	22	1,69E-07		6773,7968
	x20	2,54E-07			x20	9,21E-08		x5	1,69E-07			

Tabelle A.7: des

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	8	1,21E-07	0	21	14	2,38E-07	3	16,132	15	1,17E-07	8	41,041
2	8	1,21E-07	0	20	14	2,37E-07	2	53,4	15	1,17E-07	8	38,651
3	8	1,21E-07	0	20	14	2,32E-07	2	51,459	15	1,17E-07	8	40,702
4	8	1,21E-07	0	20	14	2,34E-07	2	53,441	15	1,17E-07	8	38,813
5	8	1,21E-07	0	20	14	2,34E-07	3	14,792	15	1,17E-07	8	40,237
6	8	1,21E-07	0	20	14	1,46E-07	2	57,304				
7	8	1,21E-07	0	21	14	2,12E-07	2	53,552				
8	8	1,21E-07	0	21	14	2,13E-07	2	57,787				
9	8	1,21E-07	0	20	14	2,29E-07	2	47,395				
10	8	1,21E-07	0	21	14	2,37E-07	3	1,679				
11	8	1,21E-07	0	20	14	2,40E-07	2	55,168				
12	8	1,21E-07	0	21	13	2,35E-07	2	34,615				
13	8	1,21E-07	0	22	14	2,41E-07	2	51,45				
14	8	1,21E-07	0	23	14	2,39E-07	2	57,75				
15	8	1,21E-07	0	22	14	2,02E-07	3	14,654				
16	8	1,21E-07	0	21	14	1,57E-07	2	56,672				
17	8	1,21E-07	0	21	14	2,08E-07	2	49,268				
18	8	1,21E-07	0	21	14	2,29E-07	3	14,978				
19	8	1,21E-07	0	21	13	1,40E-07	2	4,326				
20	8	1,21E-07	0	21	14	2,11E-07	3	20,211				
	8	1,21E-07	0	21	13,9	2,16E-07	2,3	38,30165	15	1,17E-07	8	39,8888
	x20	1,21E-07		21	13	2,41E-07		176,30165	15	1,17E-07		519,8888
				x2	1,40E-07		x5	1,17E-07				

Tabelle A.8: diffeq

#	vpr430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	8	8,82E-08	0	9,757	14	1,05E-08	1	22,232	14 1,10E-08
2	8	8,82E-08	0	9,868	13	1,87E-08	1	20,104	14 1,10E-08
3	8	8,82E-08	0	10,056	14	1,42E-08	1	24,678	14 1,10E-08
4	8	8,82E-08	0	9,514	13	1,05E-08	0	43,643	14 1,10E-08
5	8	8,82E-08	0	9,423	14	1,05E-08	1	22,443	14 1,10E-08
6	8	8,82E-08	0	9,78	14	1,10E-08	1	22,443	
7	8	8,82E-08	0	9,417	14	1,05E-08	1	22,557	
8	8	8,82E-08	0	9,523	14	1,05E-08	1	21,921	
9	8	8,82E-08	0	9,64	14	1,14E-08	1	25,31	
10	8	8,82E-08	0	9,613	14	1,05E-08	1	33,662	
11	8	8,82E-08	0	9,638	14	1,10E-08	1	44,855	
12	8	8,82E-08	0	9,487	14	1,10E-08	1	22,578	
13	8	8,82E-08	0	9,663	14	1,05E-08	1	30,244	
14	8	8,82E-08	0	9,659	14	1,05E-08	1	21,512	
15	8	8,82E-08	0	9,518	14	1,10E-08	1	23,264	
16	8	8,82E-08	0	9,525	13	1,05E-08	1	22,551	
17	8	8,82E-08	0	9,604	14	1,10E-08	1	21,79	
18	8	8,82E-08	0	9,474	14	1,10E-08	1	23,658	
19	8	8,82E-08	0	10,244	14	1,05E-08	1	25	
20	8	8,82E-08	0	9,89	13	1,05E-08	1	3,564	
	8	8,82E-08	0	9,66465	13,8	1,13E-08	0,95	24,90045	14 1,10E-08
	x20	8,82E-08		9,66465	13	1,87E-08		81,90045	14 1,10E-08
	8	8,82E-08		x4	1,05E-08		x5	1,10E-08	133,5402

Tabelle A.9: dsip

#	chan_width	vpr430			parallel lee			sequentiell lee			
		crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)
1	6	7,98E-08	2	41,101	10	5,16E-08	1	54,996	11	5,07E-08	4
2	6	7,98E-08	2	41,848	10	5,16E-08	2	4,080	11	5,07E-08	4
3	6	7,98E-08	2	40,947	10	5,16E-08	2	12,806	11	5,07E-08	4
4	6	7,98E-08	2	41,376	10	5,16E-08	1	55,167	11	5,07E-08	4
5	6	7,98E-08	2	42,562	10	5,16E-08	2	2,672	11	5,07E-08	4
6	6	7,98E-08	2	41,744	10	5,16E-08	2	14,678			
7	6	7,98E-08	2	42,095	10	5,16E-08	2	9,145			
8	6	7,98E-08	2	40,831	10	5,16E-08	2	39,016			
9	6	7,98E-08	2	42,215	10	9,31E-08	2	21,433			
10	6	7,98E-08	2	41,242	10	5,16E-08	2	4,398			
11	6	7,98E-08	2	41,753	10	5,16E-08	2	17,725			
12	6	7,98E-08	2	42,887	10	5,75E-08	2	9,801			
13	6	7,98E-08	2	41,071	10	5,16E-08	1	56,636			
14	6	7,98E-08	2	41,664	10	7,94E-08	2	32,040			
15	6	7,98E-08	2	40,889	10	8,67E-08	1	58,113			
16	6	7,98E-08	2	40,461	10	5,16E-08	2	21,927			
17	6	7,98E-08	2	42,061	10	5,16E-08	2	12,789			
18	6	7,98E-08	2	41,532	10	5,39E-08	1	55,451			
19	6	7,98E-08	2	41,953	10	5,16E-08	1	54,575			
20	6	7,98E-08	2	41,951	10	5,16E-08	1	59,710			
	6	7,98E-08	2	41,60915	10	5,72E-08	1,65	29,858	11	5,07E-08	4
	x20	7,98E-08		161,60915	10	9,31E-08		128,858	11	5,07E-08	281,222
	x20	7,98E-08			x20	5,16E-08		x5	5,07E-08		

Tabelle A.10: elliptic

#	vpr430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	11	1,83E-07	2	21,971	18	1,51E-08	12	51,916	20
2	11	1,83E-07	2	22,607	18	1,29E-08	13	15,028	20
3	11	1,83E-07	2	22,382	18	2,24E-08	12	47,632	20
4	11	1,83E-07	2	23,557	18	1,64E-08	12	17,097	20
5	11	1,83E-07	2	23,246	18	1,29E-08	12	45,966	20
6	11	1,83E-07	2	24,645	18	1,29E-08	13	55,266	20
7	11	1,83E-07	2	23,136	18	1,51E-08	12	20,747	20
8	11	1,83E-07	2	22,951	18	1,29E-08	12	28,72	20
9	11	1,83E-07	2	22,546	18	1,29E-08	13	7,537	20
10	11	1,83E-07	2	23,14	18	1,87E-08	12	57,124	20
11	11	1,83E-07	2	22,923	18	1,33E-08	12	23,542	20
12	11	1,83E-07	2	21,738	18	1,29E-08	13	41,469	20
13	11	1,83E-07	2	22,749	18	1,46E-08	12	57,943	20
14	11	1,83E-07	2	22,424	18	1,60E-08	14	18,734	20
15	11	1,83E-07	2	22,851	18	1,60E-08	12	30,821	20
16	11	1,83E-07	2	22,267	18	1,29E-08	12	56,674	20
17	11	1,83E-07	2	22,505	18	1,29E-08	13	16,521	20
18	11	1,83E-07	2	23,07	18	1,69E-08	12	24,29	20
19	11	1,83E-07	2	22,964	18	1,29E-08	12	35,402	20
20	11	1,83E-07	2	23,204	18	1,60E-08	12	20,312	20
	11	1,83E-07	2	22,8438	18	1,48E-08	12,35	33,63705	20
	x20	1,83E-07		142,8438	18	2,24E-08		774,63705	20
	x20	1,83E-07			x20	1,29E-08		x5	1,64E-08
	11	1,83E-07							1216,5156

Tabelle A.11: ex5p

#	vpr430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	14	1,09E-07	0	11,114	21	1,42E-07	1	20,231	21
2	14	1,09E-07	0	11,201	20	1,25E-07	1	22,642	21
3	14	1,09E-07	0	11,115	20	1,23E-07	1	36,32	21
4	14	1,09E-07	0	11,116	20	1,29E-07	1	20,003	21
5	14	1,09E-07	0	11,188	20	1,25E-07	1	21,879	21
6	14	1,09E-07	0	11,189	20	1,28E-07	1	37,758	
7	14	1,09E-07	0	11,196	20	1,23E-07	1	22,407	
8	14	1,09E-07	0	11,222	20	1,27E-07	1	20,329	
9	14	1,09E-07	0	11,193	20	1,49E-07	1	30,849	
10	14	1,09E-07	0	11,259	20	1,35E-07	1	28,664	
11	14	1,09E-07	0	11,139	20	1,28E-07	1	45,905	
12	14	1,09E-07	0	11,607	20	1,25E-07	1	23,224	
13	14	1,09E-07	0	11,269	20	1,30E-07	1	27,088	
14	14	1,09E-07	0	11,04	20	1,30E-07	1	44,813	
15	14	1,09E-07	0	11,285	20	1,46E-07	1	20,193	
16	14	1,09E-07	0	11,141	20	1,17E-07	1	22,61	
17	14	1,09E-07	0	11,165	20	1,33E-07	1	26,906	
18	14	1,09E-07	0	11,004	20	1,31E-07	1	18,421	
19	14	1,09E-07	0	11,262	20	1,13E-07	1	26,866	
20	14	1,09E-07	0	11,272	20	1,30E-07	1	28,117	
	14	1,09E-07	0	11,19885	20,05	1,29E-07	1	27,26125	21
	x19	1,09E-07		x19	1,13E-07			1,31E-07	2
	14	1,09E-07		11,19885	20	1,49E-07		87,26125	21
	x20	1,09E-07			x5	1,31E-07			134,0234

Tabelle A.12: ex1010

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	11	2,48E-07	4	20,791	18	2,85E-07	15	37,798	18	2,66E-07	27	17,802
2	11	2,48E-07	4	22,978	18	3,13E-07	15	24,347	18	2,66E-07	27	24,004
3	11	2,48E-07	4	19,876	17	3,02E-07	13	22,632	18	2,66E-07	27	19,364
4	11	2,48E-07	4	19,957	18	2,93E-07	15	13,549	18	2,66E-07	27	18,202
5	11	2,48E-07	4	19,993	18	2,84E-07	15	6,91	18	2,66E-07	27	17,222
6	11	2,48E-07	4	21,352	18	2,84E-07	15	32,254				
7	11	2,48E-07	4	19,671	17	2,88E-07	10	49,482				
8	11	2,48E-07	4	22,443	18	3,32E-07	15	26,712				
9	11	2,48E-07	4	20,046	17	2,80E-07	11	26,653				
10	11	2,48E-07	4	20,909	17	3,02E-07	13	41,03				
11	11	2,48E-07	4	20,854	18	2,88E-07	15	47,276				
12	11	2,48E-07	4	22,151	18	2,56E-07	15	6,472				
13	11	2,48E-07	4	42,913	17	2,95E-07	10	40,476				
14	11	2,48E-07	4	24,397	17	2,97E-07	11	29,637				
15	11	2,48E-07	4	22,438	18	2,62E-07	15	23,808				
16	11	2,48E-07	4	20,139	18	3,37E-07	15	8,449				
17	11	2,48E-07	4	22,791	17	2,96E-07	11	6,406				
18	11	2,48E-07	4	24,169	17	2,89E-07	13	14,004				
19	11	2,48E-07	4	22,808	18	2,94E-07	15	16,781				
20	11	2,48E-07	4	21,078	18	3,26E-07	15	31,676				
	11	2,48E-07	4	22,5877	17,6	2,95E-07	13,6	25,3176	18	2,66E-07	27	19,3188
x20	11	2,48E-07		262,5877	17	3,37E-07		841,3176	18	2,66E-07		1639,3188
x20					x8	2,56E-07		x5		2,66E-07		

Tabelle A.13: frisc

#	vpr430			parallel lee			sequentiell lee					
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)			
1	13	1,69E-07	0	26,02	20	6,44E-08	8	28,908	20	6,16E-08	17	19,578
2	13	1,69E-07	0	26,028	20	6,71E-08	8	41,647	20	6,16E-08	17	47,644
3	13	1,69E-07	0	26,538	20	4,47E-08	8	22,89	20	6,16E-08	18	25,204
4	13	1,69E-07	0	26,18	19	6,61E-08	5	40,976	20	6,16E-08	17	24,202
5	13	1,69E-07	0	26	20	6,44E-08	8	22,294	20	6,16E-08	17	50,309
6	13	1,69E-07	0	26,295	20	5,98E-08	8	23,56				
7	13	1,69E-07	0	26,022	20	6,44E-08	8	14,309				
8	13	1,69E-07	0	26,168	20	6,07E-08	8	22,102				
9	13	1,69E-07	0	26,503	20	6,07E-08	9	3,711				
10	13	1,69E-07	0	26,617	20	6,44E-08	8	18,166				
11	13	1,69E-07	0	26,338	20	6,62E-08	8	18,416				
12	13	1,69E-07	0	25,727	20	6,61E-08	8	26,803				
13	13	1,69E-07	0	25,887	19	6,29E-08	5	45,464				
14	13	1,69E-07	0	25,902	19	6,53E-08	7	9,632				
15	13	1,69E-07	0	26,4	20	6,44E-08	8	20,71				
16	13	1,69E-07	0	26,022	20	6,53E-08	8	18,02				
17	13	1,69E-07	0	26,011	20	6,53E-08	8	8,738				
18	13	1,69E-07	0	26,092	20	6,44E-08	8	10,922				
19	13	1,69E-07	0	26,059	20	8,39E-08	8	16,835				
20	13	1,69E-07	0	26,076	20	6,07E-08	8	34,186				
	13	1,69E-07	0	26,14425	19,85	6,41E-08	7,7	22,41445	20	6,16E-08	17,2	33,3874
x20	1,69E-07			26,14425	19	8,39E-08		484,41445	20	6,16E-08		1065,3874
					x3	4,47E-08			x5	6,16E-08		

Tabelle A.14: misex3

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	12	1,06E-07	0	52,672	18	1,38E-07	2	7,356	18	1,33E-07	3	32,669
2	12	1,06E-07	0	52,204	18	1,45E-07	2	9,726	18	1,33E-07	3	33,46
3	12	1,06E-07	0	52,433	18	1,20E-07	2	14,257	18	1,33E-07	3	32,992
4	12	1,06E-07	0	52,212	18	1,18E-07	2	10,131	18	1,33E-07	3	33,205
5	12	1,06E-07	0	52,553	18	1,24E-07	2	8,072	18	1,33E-07	3	33,375
6	12	1,06E-07	0	52,457	18	1,24E-07	2	9,464				
7	12	1,06E-07	0	52,873	18	1,35E-07	2	9,765				
8	12	1,06E-07	0	52,853	18	1,19E-07	2	7,861				
9	12	1,06E-07	0	52,112	18	1,19E-07	2	11,669				
10	12	1,06E-07	0	52,508	18	1,20E-07	2	9,675				
11	12	1,06E-07	0	52,158	18	1,32E-07	2	9,504				
12	12	1,06E-07	0	52,312	18	1,33E-07	2	9,628				
13	12	1,06E-07	0	52,262	18	1,37E-07	2	8,767				
14	12	1,06E-07	0	52,274	18	1,30E-07	2	14,081				
15	12	1,06E-07	0	52,217	18	1,23E-07	2	13,195				
16	12	1,06E-07	0	52,741	18	1,28E-07	2	8,291				
17	12	1,06E-07	0	52,81	18	1,39E-07	2	10,021				
18	12	1,06E-07	0	52,322	18	1,31E-07	2	9,529				
19	12	1,06E-07	0	52,465	18	1,34E-07	2	8,912				
20	12	1,06E-07	0	51,768	18	1,23E-07	2	10,301				
				52,4103	18	1,29E-07	2	10,01025	18	1,33E-07	3	33,1402
	x20	1,06E-07			x20	1,18E-07				x5	1,33E-07	213,1402
	12	1,06E-07										

Tabelle A.15: pdc

#	vpf430			parallel lee			sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	18	2,39E-07	5	54,443	26	2,65E-07	20	40,939	26
2	18	2,39E-07	5	52,902	26	2,67E-07	20	20,857	26
3	18	2,39E-07	5	52,828	26	3,28E-07	20	42,578	26
4	18	2,39E-07	5	52,251	26	2,84E-07	20	28,602	26
5	18	2,39E-07	5	50,528	26	2,88E-07	20	22,6	26
6	18	2,39E-07	5	54,324	26	2,64E-07	20	24,044	
7	18	2,39E-07	5	50,283	26	2,93E-07	21	12,657	
8	18	2,39E-07	5	52,311	26	3,00E-07	20	56,764	
9	18	2,39E-07	5	52,451	26	3,07E-07	20	47,929	
10	18	2,39E-07	5	53,576	26	2,83E-07	21	7,679	
11	18	2,39E-07	5	48,219	26	2,75E-07	20	44,897	
12	18	2,39E-07	5	51,195	26	2,87E-07	20	31,609	
13	18	2,39E-07	5	51,851	26	2,55E-07	21	49,002	
14	18	2,39E-07	5	50,486	26	2,58E-07	21	8,015	
15	18	2,39E-07	5	51,582	26	2,83E-07	20	33,532	
16	18	2,39E-07	5	52,136	26	2,73E-07	20	42,205	
17	18	2,39E-07	5	52,445	26	2,68E-07	20	49,289	
18	18	2,39E-07	5	51,482	26	2,67E-07	20	43,55	
19	18	2,39E-07	5	52,269	26	3,99E-07	20	40,957	
20	18	2,39E-07	5	51,906	26	2,75E-07	20	45,65	
	18	2,39E-07	5	51,9734	26	2,83E-07	20,2	34,66775	26
	x20	2,39E-07		351,9734	26	3,49E-07		1246,66775	26
	x20	2,39E-07			x20	2,55E-07		x5	2,90E-07
	18	2,39E-07							2670,7008

Tabelle A.16: s298

#	vpi430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	8	1,98E-07	0	28,288	13	1,26E-07	1	34,732	14	1,14E-07	3	2,253
2	8	1,98E-07	0	27,637	13	1,39E-07	1	55,567	14	1,14E-07	3	1,555
3	8	1,98E-07	0	27,812	13	1,01E-07	1	44,206	14	1,14E-07	3	1,983
4	8	1,98E-07	0	28,019	13	1,40E-07	1	9,611	14	1,14E-07	3	2,396
5	8	1,98E-07	0	27,473	13	1,21E-07	1	9,472	14	1,14E-07	3	3,568
6	8	1,98E-07	0	27,831	13	1,01E-07	2	0,544				
7	8	1,98E-07	0	27,81	14	1,35E-07	2	7,218				
8	8	1,98E-07	0	29,23	13	1,25E-07	1	31,022				
9	8	1,98E-07	0	27,949	13	9,58E-08	1	35,334				
10	8	1,98E-07	0	29,802	13	9,92E-08	1	9,339				
11	8	1,98E-07	0	27,65	14	1,15E-07	2	8,667				
12	8	1,98E-07	0	27,648	14	1,13E-07	2	7,886				
13	8	1,98E-07	0	27,804	13	1,57E-07	1	22,468				
14	8	1,98E-07	0	27,954	13	9,46E-08	1	11,061				
15	8	1,98E-07	0	27,494	13	1,27E-07	1	30,935				
16	8	1,98E-07	0	28,041	13	1,19E-07	1	20,321				
17	8	1,98E-07	0	27,539	13	1,31E-07	1	18,334				
18	8	1,98E-07	0	27,718	13	1,13E-07	1	41,106				
19	8	1,98E-07	0	28,073	13	1,11E-07	1	13,996				
20	8	1,98E-07	0	27,779	13	1,36E-07	1	48,68				
	8	1,98E-07	0	27,97755	13,15	1,20E-07	1,2	23,02495	14	1,14E-07	3	2,351
	x20	1,98E-07		27,97755	13	1,57E-07		95,02495	14	1,14E-07		182,351
				x17	9,46E-08		x5	1,14E-07				

Tabelle A.17: s38417

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	8	1,77E-07	0	53,281	15	1,14E-07	21	1,826	15	1,03E-07	41	23,697
2	8	1,77E-07	0	51,552	14	1,41E-07	19	48,986	15	1,03E-07	41	23,661
3	8	1,77E-07	0	52,251	15	9,99E-08	20	59,547	15	1,03E-07	41	26,788
4	8	1,77E-07	0	51,343	14	1,31E-07	14	19,87	15	1,03E-07	41	5,729
5	8	1,77E-07	0	51,195	14	1,08E-07	20	40,443	15	1,03E-07	40	55,672
6	8	1,77E-07	0	56,373	14	1,17E-07	28	15,879				
7	8	1,77E-07	0	54,76	14	1,10E-07	15	18,228				
8	8	1,77E-07	0	54,595	14	1,18E-07	17	21,149				
9	8	1,77E-07	0	53,348	14	1,09E-07	16	1,912				
10	8	1,77E-07	0	51,866	14	1,67E-07	15	6,24				
11	8	1,77E-07	0	51,906	15	1,16E-07	20	45,201				
12	8	1,77E-07	0	51,938	14	1,25E-07	15	23,748				
13	8	1,77E-07	0	51,799	15	1,52E-07	21	4,326				
14	8	1,77E-07	0	52,885	14	1,10E-07	13	26,106				
15	8	1,77E-07	0	51,964	15	1,48E-07	21	7,505				
16	8	1,77E-07	0	51,629	15	1,14E-07	22	32,695				
17	8	1,77E-07	0	52,276	14	1,16E-07	18	54,706				
18	8	1,77E-07	0	51,975	15	1,09E-07	21	14,018				
19	8	1,77E-07	0	51,359	15	1,16E-07	21	2,878				
20	8	1,77E-07	0	51,642	14	1,18E-07	19	45,711				
	8	1,77E-07	0	52,49685	14,4	1,22E-07	18,8	24,5487	15	1,03E-07	40,8	27,1094
x20	8	1,77E-07		52,49685	14	1,67E-07		1152,5487	15	1,03E-07		2475,1094
	x20	8	1,77E-07		x12	9,99E-08		x5	1,03E-07			

Tabelle A.18: s38584.1

#	vpr430			parallel lee						sequentiell lee		
	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)	chan_width	crit_t(s)	Laufzeit(m)
1	9	1,27E-07	1	39,899	14	1,87E-07	18	8,085	15	1,89E-07	49	50,476
2	9	1,27E-07	1	39,891	14	1,76E-07	18	33,14	15	1,89E-07	51	46,091
3	9	1,27E-07	1	40,104	14	1,65E-07	17	43,881	15	1,89E-07	50	4,852
4	9	1,27E-07	1	39,588	14	1,88E-07	22	38,225	15	1,89E-07	49	54,964
5	9	1,27E-07	1	39,811	14	1,61E-07	18	34,972	15	1,89E-07	50	1,837
6	9	1,27E-07	1	39,357	14	1,74E-07	18	31,183				
7	9	1,27E-07	1	39,268	14	1,90E-07	18	29,806				
8	9	1,27E-07	1	39,507	14	2,48E-07	17	44,553				
9	9	1,27E-07	1	39,812	14	1,63E-07	19	2,016				
10	9	1,27E-07	1	39,408	14	2,06E-07	18	12,244				
11	9	1,27E-07	1	39,989	14	1,75E-07	18	8,461				
12	9	1,27E-07	1	38,377	14	1,76E-07	17	56,403				
13	9	1,27E-07	1	38,606	14	1,74E-07	18	46,614				
14	9	1,27E-07	1	39,557	14	2,03E-07	18	40,705				
15	9	1,27E-07	1	38,878	14	1,85E-07	18	22,145				
16	9	1,27E-07	1	38,471	14	1,54E-07	19	7,686				
17	9	1,27E-07	1	38,547	14	1,64E-07	18	40,728				
18	9	1,27E-07	1	39,228	14	1,80E-07	17	56,528				
19	9	1,27E-07	1	38,657	14	1,79E-07	18	37,689				
20	9	1,27E-07	1	38,372	14	1,59E-07	18	9,816				
	9	1,27E-07	1	39,26635	14	1,80E-07	18,1	30,244	15	1,89E-07	49,8	31,644
	x20	1,27E-07		99,26635	14	2,48E-07		1116,244	x5	1,89E-07		3019,644

Tabelle A.19: seq

#	vpt430					parallel lee					sequentiell lee					
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	12	1,27E-07	1	4,301	18	1,75E-07	3	4,227	19	1,73E-07	3	10,467				
2	12	1,27E-07	1	4,274	18	1,50E-07	3	25,343	19	1,73E-07	3	9,025				
3	12	1,27E-07	1	3,623	18	1,60E-07	3	3,829	19	1,73E-07	3	51,825				
4	12	1,27E-07	1	4,403	18	1,61E-07	3	39,675	19	1,73E-07	3	10,627				
5	12	1,27E-07	1	4,137	19	1,46E-07	1	37,868	19	1,73E-07	3	10,932				
6	12	1,27E-07	1	3,68	18	1,57E-07	3	11,43								
7	12	1,27E-07	1	4,042	18	1,60E-07	2	52,455								
8	12	1,27E-07	1	3,876	18	1,50E-07	2	53,333								
9	12	1,27E-07	1	4,019	18	1,52E-07	2	54,629								
10	12	1,27E-07	1	4,293	18	1,72E-07	3	4,789								
11	12	1,27E-07	1	3,826	18	1,63E-07	3	18,401								
12	12	1,27E-07	1	4,021	18	1,60E-07	3	0,213								
13	12	1,27E-07	1	3,412	18	1,56E-07	2	53,742								
14	12	1,27E-07	1	3,759	18	1,49E-07	3	13,237								
15	12	1,27E-07	1	2,961	18	1,54E-07	3	40,217								
16	12	1,27E-07	1	3,806	18	1,51E-07	3	19,006								
17	12	1,27E-07	1	3,919	18	1,60E-07	2	54,241								
18	12	1,27E-07	1	4,157	18	1,74E-07	2	49,7								
19	12	1,27E-07	1	4,194	18	1,53E-07	2	55,094								
20	12	1,27E-07	1	4,221	18	1,50E-07	3	31,203								
				3,9462	18,05	1,58E-07	2,55	31,1316	19	1,73E-07	3	18,5752				
	12	1,27E-07	x20	63,9462	18	1,75E-07	184,1316	19	1,73E-07	x5	1,73E-07	198,5752				
				x19	1,46E-07											

Tabelle A.20: spla

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	14	1,76E-07	1	17,244	22	2,69E-07	10	28,837	23	2,60E-07	14	21,365
2	14	1,76E-07	1	16,239	22	2,46E-07	10	50,479	23	2,60E-07	14	28,109
3	14	1,76E-07	1	16,939	22	2,33E-07	10	8,773	23	2,60E-07	14	38,465
4	14	1,76E-07	1	16,75	22	2,63E-07	10	42,478	23	2,60E-07	14	24,844
5	14	1,76E-07	1	16,457	22	2,39E-07	11	23,408	23	2,60E-07	14	21,156
6	14	1,76E-07	1	17,1	22	2,23E-07	10	28,608				
7	14	1,76E-07	1	16,826	22	2,54E-07	11	13,421				
8	14	1,76E-07	1	16,85	22	2,72E-07	11	6,503				
9	14	1,76E-07	1	16,336	22	2,62E-07	12	18,403				
10	14	1,76E-07	1	16,572	22	2,30E-07	10	4,918				
11	14	1,76E-07	1	17,064	23	2,27E-07	8	5,004				
12	14	1,76E-07	1	16,257	22	2,84E-07	9	55,199				
13	14	1,76E-07	1	16,479	22	2,31E-07	10	30,448				
14	14	1,76E-07	1	16,26	22	2,38E-07	10	24,064				
15	14	1,76E-07	1	16,177	22	2,36E-07	11	51,152				
16	14	1,76E-07	1	16,526	22	2,34E-07	10	20,88				
17	14	1,76E-07	1	16,193	22	2,59E-07	10	3,783				
18	14	1,76E-07	1	16,12	22	2,41E-07	10	40,104				
19	14	1,76E-07	1	15,887	22	2,25E-07	11	19,48				
20	14	1,76E-07	1	16,614	22	2,25E-07	9	53,994				
	14	1,76E-07	1	16,5445	22,05	2,45E-07	10,15	26,4968	23	2,60E-07	14	26,7878
x20	1,76E-07			76,5445	22	2,84E-07		635,4968	23	2,60E-07		866,7878
				x19	2,23E-07			x5	2,60E-07			

Tabelle A.21: tseng

#	vpr430				parallel lee				sequentiell lee			
	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)	chan_width	crit_t(s)	Laufzeit(m)	Laufzeit(s)
1	7	9,00E-08	0	2,627	12	3,03E-08	1	11,751	14	3,76E-08	1	22,688
2	7	9,00E-08	0	2,467	12	3,03E-08	1	13,55	14	3,76E-08	1	22,731
3	7	9,00E-08	0	2,446	12	3,03E-08	1	21,416	14	3,76E-08	1	22,724
4	7	9,00E-08	0	2,664	12	3,22E-08	1	15,59	14	3,76E-08	1	27,582
5	7	9,00E-08	0	2,575	12	3,45E-08	1	11,187	14	3,76E-08	1	23,036
6	7	9,00E-08	0	2,615	12	3,03E-08	1	15,376				
7	7	9,00E-08	0	2,505	12	3,03E-08	1	12,089				
8	7	9,00E-08	0	2,618	12	3,45E-08	1	12,888				
9	7	9,00E-08	0	2,512	12	3,13E-08	1	11,274				
10	7	9,00E-08	0	2,522	12	3,45E-08	1	11,367				
11	7	9,00E-08	0	2,519	12	3,63E-08	1	12,966				
12	7	9,00E-08	0	2,711	12	3,45E-08	1	11,412				
13	7	9,00E-08	0	2,524	12	3,45E-08	1	18,526				
14	7	9,00E-08	0	2,459	12	3,31E-08	1	14,598				
15	7	9,00E-08	0	2,611	12	3,03E-08	1	12,552				
16	7	9,00E-08	0	2,618	12	3,03E-08	1	14,618				
17	7	9,00E-08	0	2,58	12	3,31E-08	1	11,296				
18	7	9,00E-08	0	2,584	12	3,03E-08	1	15,023				
19	7	9,00E-08	0	2,666	12	3,03E-08	1	14,069				
20	7	9,00E-08	0	2,745	12	3,45E-08	1	13,104				
	7	9,00E-08	0	2,5784	12	3,23E-08	1	13,7326	14	3,76E-08	1	23,7522
x20	9,00E-08		2,5784		12	3,63E-08		73,7326	14	3,76E-08		83,7522
			x20	9,00E-08		x5	3,03E-08			x5	3,76E-08	

A.2 Quellcode

https://github.com/ZuzoSteppenwolf/parallel_Lee