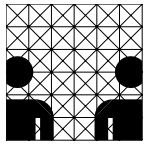


Inhaltsverzeichnis



Andreas Mäder

Universität Hamburg – Fachbereich Informatik
Arbeitsbereich Technische Aspekte Multimodaler Systeme

<http://tams-www.informatik.uni-hamburg.de>

Da die hier bereitgestellten Dateien nicht alle einzeln beschrieben werden können (ca. 2500 Stück), gibt dieses Dokument einen Überblick über den Aufbau und Inhalt der Verzeichnisse sowie die zugrunde liegende Namenskonvention.

promotion/ – die Arbeit im PDF-Format, *passwort erforderlich*
dissertation.pdf PDF-Version des Buchs, mit Hyperlinks
disputation.pdf Disputationsvortrag, als PDF-Präsentation

beispiele/

Zu dem Buch gehörende Beispieldateien mit kommentiertem, lauffähigem VHDL-Code. Der Dateiname entspricht dem Bezeichner der jeweiligen (top-level) Entity. In jeder Datei sind diese Entity mit den zugehörigen Architekturen – in der Regel genau eine, bei einigen Beispielen aber auch mehrere – und ggf. vorhandene Konfigurationen zusammengefasst. Ein Kommentar im Kopf beschreibt, welche VHDL-Einheiten vorhanden sind, wo sie im Buch stehen und welche Abhängigkeiten zu anderen VHDL-Quelldateien (Hierarchie) oder zu Eingabe-/Ausgabedateien bestehen.

benchmark/ – VHDL-Simulationsbenchmarks

results.pdf Zusammenfassung *aller* Benchmarkergebnisse als Tabellen und in graphischer Form. Die Tabellen sind so auch in der Dissertation enthalten, von den generierten Gnuplot-Graphiken wurden aber nur wenige in die Arbeit übernommen.

clock/ Benchmark: „Funkwecker“
mult/ – „Booth-Multiplizierer“
opw/ – „Operationswerk“
tlcWalk/ – „Fußgängerampel“

win/ Simulationsverzeichnisse und -Skripte für die Benchmarks auf dem Windows-PC. Allerdings wurden dort nur einige Simulationen auf Register-Transfer Ebene durchgeführt (Simulatoren: ModelSim und Simili), um einen Geschwindigkeitsvergleich mit Workstations zu erhalten.

envData/ Beschreibungen zur Simulationsumgebung: die Programmversionen, die eingesetzte Hardware, Profiling-Information zu den Schaltungsgrößen etc.

plot/ Skripte zur Datenaufbereitung für Gnuplot und LaTeX

tex/ LaTeX-Tabellen mit den Benchmarkergebnissen

Inhalte der Benchmarkverzeichnisse

Alle Verzeichnisse sind nach dem gleichen Schema aufgebaut, wobei aber je nach „Art“ der Benchmarks einige Verzeichnisse fehlen können, so zum Beispiel die Netzlistensimulationen der Fußgängerampel oder für den Multiplizierer die Synthese und Simulationen mit dem UMC-Fertigungsprozess.

Basisverzeichnisse enthalten mehrfach benutzte Dateien, auf die dann in den Simulationsverzeichnissen verwiesen wird.

src/ VHDL-Quelldateien mit synthesesfähigem RT-Code und die zugehörigen Testumgebungen, deren Namenskonvention ist: $\langle top-level \rangle Tst.vhd$

syn/ Synthetisierte Netzlisten im VHDL- und Verilog-Format: syn für AMS- und oldSyn für den UMC-Prozess (s.u.)

Simulationsverzeichnisse mit allen Eingabe- und Ausgabedateien der Benchmarks. Ihre Bezeichner setzen sich aus drei Teilen zusammen:

1 – einem Präfix zur Auswahl der Zellbibliotheken

$\langle dir \rangle /$ AMS-Prozess, wenn keine weitere Angabe
old $\langle Dir \rangle /$ UMC-Prozess

2 – einem Suffix der die Simulationsart beschreibt

$\langle dir \rangle /$ RT-Code
 $\langle dir \rangle Ref /$ RT-Referenz zum Vergleich mit den Netzlistensimulationen, wenn die Anzahl der Stimuli durch einen top-level Generic variiert wird (Operationswerk und Multiplizierer)

$\langle dir \rangle Syn /$ VHDL-Netzliste
 $\langle dir \rangle Mix /$ mixed-mode Netzlistensimulation

3 – einem Code für den Simulator, bzw. Hersteller

$\langle dir \rangle$	$\langle sim \rangle$	Simulator	mixed-mode
leapfrog	sv	CADENCE Leapfrog	(+Verilog-XL)
ncvhdl	ncsim	–”– NC Sim	
synopsys	vss vss01	SYNOPSYS VSS	
	cyc cyc01	–”– Cyclone	
	sci sci01 sci02	–”– Scirocco	(+VCS)
verilog	alle Verilog-Netzlistensimulationen		
	verilog	CADENCE Verilog-XL	
	ncsim	–”– NC Sim	
	vcs vcs62	SYNOPSYS VCS	

Die Dateinamen in den Verzeichnissen ergeben sich aus dem oben beschriebenen Code für den Simulator $\langle sim \rangle$ und den folgenden Endungen:

$\langle sim \rangle .csh$ ein Shell-Skript, das alle Simulationsläufe im Batch-Betrieb steuert
 $\langle sim \rangle .scr$ gegebenenfalls notwendige Skripte zur Steuerung des Simulators
 $\langle sim \rangle .out$ die Ausgabe der Simulation
 $\langle sim \rangle .time$ die Zeitmessungen der Simulationsläufe; zur Kennzeichnung unterschiedlicher Parametersätze wurden teilweise noch zusätzliche Kürzel benutzt

Die VHDL- oder Verilog-Dateien stehen in den Basisverzeichnissen — da die *symbolischen Links* des Unix-Dateisystems unter Windows als Dateien der Größe 0 Byte erscheinen, sind sie dann in den Simulationsverzeichnissen durch die entsprechenden Dateien zu ersetzen.

Datenauswertung — Die Auswertung der Benchmarkergebnisse, das heißt die Erzeugung der Grafiken und LaTeX-Tabellen, erfolgt weitgehend automatisch in dem jeweiligen Baum:

plot/

time.csh Skript, das den Simulationen entsprechende Unterverzeichnisse erzeugt und ein Perl-Skript startet, welches die time-Dateien der Simulation parst und zwei Ausgabedateien schreibt: $\langle sim \rangle$.plt als Eingabe für Gnuplot und $\langle sim \rangle$.tab als LaTeX-„tabular“.

rtl/ ref/ RT-Code Simulationsergebnisse

syn/ old VHDL-Netzliste —“—

mix/ old mixed-mode —“—

verilog/ Verilog —“—

gp/ Gnuplot Arbeitsverzeichnis mit (manuell) den Daten angepassten Kontrolldateien plot. $\langle Mode \rangle$ und den damit generierten graphischen Datenformaten $\langle mode \rangle$.eps und $\langle mode \rangle$.pdf

Weitere Untersuchungen finden sich darüber hinaus noch in folgenden Verzeichnissen: tlcWalk/ncGUI/ Simulationen, die sich mit der Handhabung der Benutzerschnittstelle beschäftigen, siehe Seite 141 — Stichwort: Kommandozeile oder graphische Oberfläche mit Code-Browser und Impulsdiagramm

.../plot/ zugehöriges Gnuplot Verzeichnis

mult/bo4Syn/ Für einen Radix-4 Booth-Multiplizierer, siehe Seite 148, wurden mehrere Implementationsmöglichkeiten einer Signed-Arithmetik untersucht — dies führte zu der Aussage von Seite 212: „Eigene Implementationen lohnen sich in der Regel nicht!“

.../plot/ zugehöriges Gnuplot Verzeichnis

mult/syn/plot/ Skalierungsgraphik des Multiplizierers, siehe Seite 166

opw/synopsys/plot/

Vergleich verschiedener Programmversionen der Simulationsprogramme