Verification of System-on-Chips using Genetic Evolutionary Test Techniques from a Software Applications Perspective

Adriel Cheng
B.E. (Hons)

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

THE UNIVERSITY OF ADELAIDE

Faculty of Engineering, Computer and Mathematical Sciences
School of Electrical and Electronic Engineering

September 2009

ABSTRACT

This thesis examines verification of system-on-a-chip (SoC) designs using a software applications test methodology that is enhanced by genetic evolutionary test generations and functional coverage.

The verification methodology facilitates application based testing using behavioural simulations before the chip is fabricated. The goal of the methodology is to verify commonly used real-life functionalities of the SoC earlier in the design process, so as to uncover design bugs that are considered most critical to actual SoC usages when the SoC is employed in its intended end-product. The verification methodology is based on a test building blocks approach, whereby many different components of various SoC application use-cases are extracted into building blocks, and then recomposed with other components to construct greater variety and range of test cases for verifying the SoC.

An important facet of the methodology is to address automated creation of these software application test cases in an effective and efficient manner. The goal is to maximise test coverage and hence bug detection likelihood using minimal verification resources and effort. To this end, test generations techniques employing single and multi objective genetic algorithms and evolutionary strategies are devised in this thesis. Using coverage and test size to drive test generations, test suites which are continually evolved to enhance SoC verifications are created, thereby achieving automated coverage driven verifications.

Another enhancement for test generation is to select the input test creation parameters in an analytical manner. A technique using Markov chains is developed to model and analyse the test generation method, and by doing so, test parameters can be selected to achieve desired verification characteristics and outcomes with greater likelihood.

To quantify verification effectiveness, a functional coverage method is formulated. The coverage method monitors attributes of the SoC design during testing. The combinations of attribute values indicate the application functionalities carried out. To address the coverage space explosion phenomenon for such combinatorial methods and facilitate the coverage measurement process, partial order domains and trajectory checking techniques from the formal verification field of symbolic trajectory evaluation are adopted.

The contributions of this thesis are a verification platform and associated tool-suite that incorporates the software applications test methodology, algorithmic test generation, and functional coverage techniques.

DECLARATION

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution to Adriel Cheng and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue, the Australasian Digital Theses Program (ADTP) and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Adriel Cheng

ACKNOWLEDGEMENTS

This thesis would not be possible without the support and contributions of many individuals. First and foremost, I express my utmost gratitude to my supervisor, Professor Cheng-Chew Lim. His influence on my research and helping to bring this thesis to fruition has been invaluable. I thank him for giving me the freedom to pursue my own ideas, challenging me to examine my research from different perspectives, and encouraging me to explore various technical domains not immediately obvious. The multi-objective optimisation and Markov modelling aspects of my research were both inspired by his insightfulness to extend my semiconductor verification work with other areas of his research. I am also thankful for the extra funding he has provided toward my research studies, conference travels, and making available computing resources to run my simulations.

The School of Electrical and Electronic Engineering in the University of Adelaide, where I have spent countless hours during past years, has provided a highly supportive and friendly environment to conduct my research. In particular, I am grateful to Professor Michael Liebelt, the head of our school. Mike has supported me in all my endeavours to apply for various sources of financial grants, overseas research exchange studies, and collaborations with industry. To the other staff members of the school, I thank them for their technical and administrative support. My fellow post-graduate colleagues have also enhanced my research experience and made my studies less stressful. They include, amongst others, Hong-Gunn Chew, Brian Ng, Kiet To, and in the earlier days, Nigel Brine and Leo Lee. I thank Kiet for all the technical assistance he has provided over the years, and the useful feedback he has given regarding my research at all my seminars.

Another person who was highly instrumental in my research is Dr. Atanas Parashkevov. During the early stages of my studies, his encouragement and advice was invaluable as I embarked on my research journey. Ata helped lay the foundations of my research investigations and provided insightful recommendations in steering my research goals; which would have a lasting influence throughout my research studies. I thank him for all his help and count myself fortunate to still be able to learn from and work with him today.

I would also like to acknowledge Dr. Jay Yantchev, who was the general manager of Motorola and then Freescale Semiconductors where I worked when I started my research studies. Jay allowed me to fulfil my research aspirations whilst continuing to work part time. I was able to maintain exposure to industry engineering projects and bring different perspectives into my research. Both Jay and Ata, and Cheng-Chew and Mike were highly supportive of combining my research studies with focus from real-life engineering projects and problems; which led to collaborations and successful application of a research government grant.

My six months of research exchange at the Institute of Microelectronics at Tsinghua University in Beijing was one of the most memorable experiences during my research studies. My time at Tsinghua University enhanced my research abilities and added another facet to my research investigations. For this, I have Professor Yihe Sun, Dr. Huho He, Ting Lei, and Zhixiong Zhou to thank for being such receptive hosts and enthusiastic research collaborators.

I also acknowledge the Australian Postgraduate Award, Australia Research Council Linkage Project Industry Grant, Endeavour Australia Cheung Kong Research Exchange Award, and other engineering institution and conference grants for funding my research studies and overseas research travels.

To Joanne, my partner in life, who has never stopped supporting and encouraging me during the writing of this thesis, I will forever be grateful to you, and look forward to our future together.

Last but not least, to my parents, whose constant support and patience throughout my research studies I am so thankful for. I am indebted to them for their love and the sacrifices they have made allowing me to pursue my research studies, and make possible the many opportunities in life that one could ever hope for.

Contents

Acronyms

API	Application programming interface
ATPG	Automatic test pattern generation
CDV	Coverage driven verification
CPU	Central processing unit, microprocessor
DFT	Discrete Fourier transform
DMA	Direct memory access transfer device
DSP	Digital signal processing
FSM	Finite state machine
GEA	Genetic evolutionary algorithms
HDL	Hardware design language
PIO	Parallel input output port device
RAM	Random access read write memory
ROM	Read-only memory
RTL	Register transfer level
SAGETEG	Software application level verification methodology genetic evolutionary test generator
SALVEM	Software application level verification methodology
SoC	System-on-chip
SRAM	static random access memory
STE	Symbolic trajectory evaluation
UART	Universal asynchronous receive transmit device

Tables of symbols

Table of symbols for Chapter 4	
Symbol	**Meaning**
ϕ	Variation operator weight adjustment factor, the amount which varies to alter variation weights each time.
γ	Variation operator success ratio.
λ	Number of tests in the children test suite population.
μ	Number of tests in the parent test suite population.
$\omega(z)$	Variation operator weight value at evolution z.
A	A matrix whose elements are a count of the number of times a specific test coverage event e is exercised by a specific snippet s.
Add	The addition variation operator.
B	A single column matrix whose row elements are a count of the number of times a specific test coverage event is exercised by a test.
C	The maximum number of snippets that can be inserted into a test.
D	The domain set of parameter values that can be chosen for a parameter.
E	The set of measurable coverage test events.

$F(z)$	A function that evaluates the average coverage fitness of tests from the combined parent and children test population $P_{\mu+\lambda}(z)$ at evolution z.
G	The target coverage fitness goal value of the test generation, above which the test generation process can be terminated.
K	The number of prior consecutive evolutions to check for population fitness improvement, such that the test generation process can be terminated if there were no improvements for the last K evolutions.
L	Set of constraints for a snippet.
M	Set of dependencies for a snippet.
Mut	The mutation variation operator.
P	Set of test programs in the test suite population.
$P_{\mu+\lambda}(z)$	The combined parent and children test suite population at evolution z.
$P_{\mu}(z)$	The parent test suite population at evolution z.
$P_{\lambda}(z)$	The children test suite population at evolution z.
$Recomb$	The recombination variation operator.
Rep	The replacement variation operator.
S	Set of snippets representing the snippets library.
Sub	The subtraction variation operator.
$TourSel_k$	The tournament selection operation that conducts a tournament between k tests, whereby the winning test selected provides highest coverage fitness.
V	Set of parameters for a snippet.
a	The tuple of measurable coverage test events that can be tested.
c	The coverage metric fitness result value.
e	A measurable coverage test event.
$f(t)$	A function that evaluates the coverage metric fitness value c for a test t.
g	A constraint or dependency check for a snippet variation operation.
g_{exp}	Explicit constraint check for a snippet variation operation.
g_{imp}	Implicit constraint check for a snippet variation operation.
$g_{pre\text{-}strict}$	Pre-strict dependency check for a snippet variation operation.
$g_{pre\text{-}non\text{-}strict}$	Pre-non-strict dependency check for a snippet variation operation.
$g_{post\text{-}strict}$	Post-strict dependency check for a snippet variation operation.
$g_{post\text{-}non\text{-}strict}$	Post-non-strict dependency check for a snippet variation operation.
$h(e, s)$	A function that evaluates the number of times a test coverage event e is exercised by a snippet s.
m	The number of measurable coverage test events.
n	Number of snippets in a test program individual.
r	Desired variation success ratio goal value, typically 1/5.
s	A snippet from the snippets library set S.
t	A test program from the test suite population.
$u(s)$	The function that selects new parameter values for a snippet s during mutation variation operation.
v	A parameter of a snippet from its set of parameters V.
x	A parameter value for a parameter from the set of parameter V.

Table of symbols for Chapter 5	
Symbol	**Meaning**
ε	Variation stability state transition probability factor. The percentage factor between 0.1 to 0.5 which is applied to the variation weight variable ω, that determines the transition probability of absorption into the variation stability state *S*.
ϕ	Variation operator weight adjustment change factor. The amount to alter variation weights each time based on closeness distance of variation success ratio to the target variation success ratio.
λ	Children test population size. The number of tests for the children test population.
μ	Parent test population size. The number of tests for the parent test population.
σ	Variation change factor. The amount by which variation usage weights are adjusted each time according to self-adaptation.
ω	Variation weight variable. The variation usage weight variable that influences the rate and amount of application of the variation operation.
A	Variation increment state. Represents the state whereby variation usage is being incremented by addition onto variation usage weights.
ABS	Symbolic representation of absorbing states in an absorbing Markov chain.
C	Concurrent snippet state. Represents the state whereby snippets that execute concurrent SoC test operations are introduced into a test.
D	Variation decrement state. Represents the state whereby variation usage is being decremented.
E	Lower variation usage intermediary state. Represents the state whereby variation usage weight is between the minimum low usage limit *L* and the 1/5 success goal ratio *G*.
F	Fixed snippet composition state. Represents the state whereby the test's snippet composition is fixed because no new or different snippets have been introduced.
G	Variation success goal state. Represents the state whereby 1/5 variation success ratio goal is achieved.
H	Upper variation usage intermediary state. Represents the state whereby variation usage weight is between the maximum upper usage limit *U* and the 1/5 success goal ratio *G*.
I	The identity matrix used as a sub matrix in the transition matrix of an absorbing Markov chain, and represents the absorptive state transitions.
J	Primarily parent tests selection state. Represents the state whereby half the test population consists predominately of a selection of parent tests, with only a few children tests.
K	The number of consecutive evolutions to check for non-fitness improvements beyond which the genetic evolutionary process can be terminated.
L	Lower variation usage limit state. Represents the state whereby the limiting lowest minimum variation usage is being applied.
M	Mean first passage time matrix. The matrix whose *ij*-th element represents the time (number of state transitions) required to transit from state *i* to state *j* for the very first time in an Ergodic Markov chain.
N	Absorbing Markov chain fundamental matrix. The matrix whose elements report the expected number of times an absorbing Markov chain will operate in each transient state.
O	One single thread execution snippet state. Represents the state whereby snippets that execute one single execution threaded SoC test operations are introduced into a test.
P	Transition matrix. A matrix whose elements correspond to transition probabilities of a Markov chain.
P^n	Long term transition matrix. The transition matrix demonstrating long term behaviour of a Markov chain, as the number of evolutionary cycles *n* approaches infinity.
Q	Transient state sub matrix. A sub matrix of the transition matrix whose elements corresponds to transition probabilities between transient states of an absorbing Markov chain.
R	Absorbing state sub matrix. A sub matrix of the transition matrix whose elements corresponds to

	transition probabilities from transient states to absorbing states of an absorbing Markov chain.
S	Variation stability state. Represents the state whereby variation usage does not change when variation success ratio has achieved desired 1/5 rate.
TR	Transient states. Symbolic representation of transient states in an absorbing Markov chain.
U	Upper variation usage limit state. Represents the state whereby the limiting highest maximum variation usage is being applied.
V	New children tests selection only state. Represents the state whereby at least half the test population consists of selection of new children tests.
X	Greater new children tests selection state. Represents the state whereby half the test population consists of greater selection of new children tests than parent tests.
Y	Less or equivalent new children tests selection state. Represents the state whereby half the test population consists of less or equal selection of new children tests than parent tests.
Z	Ergodic Markov chain fundamental matrix. The fundamental matrix of an Ergodic Markov chain, which shares similar purpose to the fundamental matrix of the absorbing Markov chain *N*.
c	Variation usage incrementation or decrementation change variable. The number of times a variation usage weight variable undergoes continual increment or decrement.
k	The number of variation usage weight increases that is permissible before maximum limit is reached.
p	Snippet selection probability. The probability of a snippet selected for use by a variation operation.
s	Variation usage switching. The number of times and proportionality a variation usage weight variable switches between incrementation and decrementation.
t	Time to absorption. The expected overall number of times an absorbing Markov chain will transition within transient states before being absorbed into the absorbing state.
w	Ergodic Markov chain row vector. The row vector whose elements report the proportion of time spent within each state of an Ergodic Markov chain.

Table of symbols for Chapter 6

Symbol	Meaning
λ	Number of tests in the children test suite population.
μ	Number of tests in the parent test suite population.
B_L	The objective subset bin of Pareto and aggregate sorted tests, sorted for the line coverage and test size objective subset *L*.
B_T	The objective subset bin of Pareto and aggregate sorted tests, sorted for the toggle coverage and test size objective subset *T*.
B_C	The objective subset bin of Pareto and aggregate sorted tests, sorted for the conditional coverage and test size objective subset *C*.
C	Subset of conditional coverage maximisation objective and test size minimisation objective.
D	The distance threshold value compared against the distance between the best and worst Pareto front plot, below which GEA termination is triggered.
G	The slope threshold value compared against the slope of consecutive test data points on a Pareto front plot, below which GEA termination is triggered.
L	Subset of line coverage maximisation objective and test size minimisation objective.
M	The maximum size permissible for a test in the SALVEM test platform.
S	Set of snippets representing the snippets library.
T	Subset of toggle coverage maximisation objective and test size minimisation objective.
f	An objective fitness function that evaluates the fitness value of an objective from an individual test.

f_a	Aggregated objective fitness function combining line coverage maximisation, toggle coverage maximisation, conditional coverage maximisation, and test size minimisation objectives fitness measures into a single fitness value, and used for aggregated ranking.
f_c	Objective fitness function for measuring and maximising conditional coverage.
f_l	Objective fitness function for measuring and maximising line coverage.
f_s	Objective fitness function for measuring and minimising test size.
f_t	Objective fitness function for measuring and maximising toggle coverage.
x	A test program individual which is a solution point in the test generation solution space.

Table of symbols for Chapter 7	
Symbol	**Meaning**
≤	Partially ordered domain relation comparison operator. Relational operator employed to compare the attribute information content captured by two domains, and their relative ordering between each other in the partial order structure. It establishes whether a domain contains greater than or equal ranges of attribute domain values when compared to another domain.
δ	*an→cq* simplify conversion function, which simplifies the formalised antecedent and consequent specifications into an equivalent attribute combinatorial set of what values each attribute should hold.
BCM	Best combinations count metric. Quantitative coverage metric for attribute combinatorial coverage. It is the count of the number of attribute combinations exercised, which are identified as unique because these combinations have not been previously realised when compared to the most recent set of exercised combinations.
CMB	A set of attribute combinations of *cmb*.
LUB	Least upper bound operator. Set theory operator that operates on partially ordered domains and is defined in Definition 7.4 Chapter 7.
STE	Symbolic trajectory evaluation, which is formal verification technique from which the control graph and partially ordered domain abstraction techniques employed for coverage measuring originate from.
WCM	Worst combinations count metric. Quantitative coverage metric for attribute combinatorial coverage. It is the count of the number of attribute combinations exercised, which are identified as unique due to its realisation of at least one unique attribute value.
a	Designates an attribute of a SoC design for reporting the types and range of SoC operations conducted.
an→cq	Antecedent and consequent specification. Formalised specifications of the attribute values that snippets use to trigger and report the status of SoC operations exercised.
cmb	Attribute combinatorial set. A tuple of attributes defined to capture and report certain behaviours of SoC operations exercised. Each attribute contain attribute values or domain variable values. The combinations of different values for each attribute in the tuples forms the attribute combinatorial set.
cumu_comb	Cumulative combinatorial set. An attribute combinatorial set that contains up to date domain values exercised thus far for each attribute in the combination set during test and coverage measure.
d	Domain variable. A variable symbol defined to capture a group of values sharing similar meaning and intention for the attribute.
f	Antecedent or consequent specification element. Simple predicate or conjunctive specification element that denotes the domain value an attribute should hold, or conjunction of attributes that should hold certain domain values.
targ_comb	Target combinatorial set. An attribute combinatorial set that contains domain values for each attribute in the set that is to be realised during testing and coverage measure. The targ_comb acts as coverage test goals whereby attribute values covered by domains given in the targ_comb must be exercised during testing.