Van atomaire gedragsbeschrijvingen naar Verilog RTL of C++/SystemC

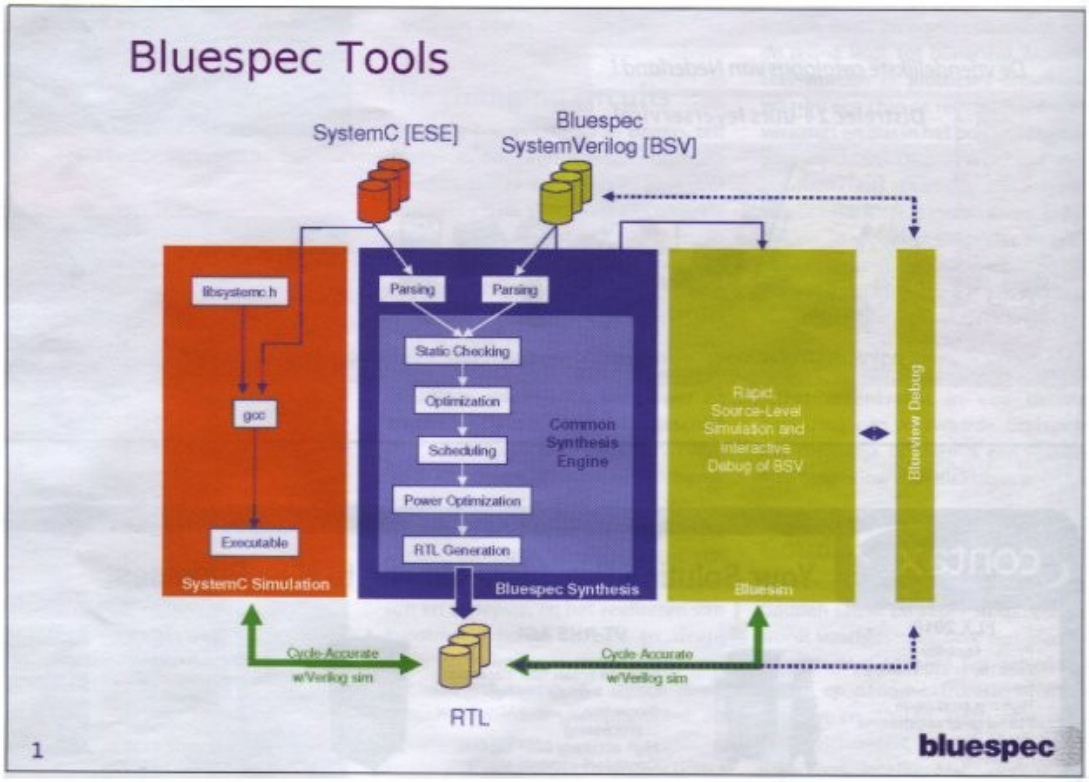
Ontwerpen met Bluespec

Het Amerikaanse bedrijf Bluespec bouwt EDA-tools voor 'electronic system level' ontwerpen. Ontwerpbeschrijvingen van atomaire transacties, zonder scheduling of timing, worden automatisch omgezet naar een parallel executeerbaar programma. Vervolgens zijn er verschillende paden naar synthese en/of simulatie. Bluespec claimt een 50% kortere ontwerptijd voor een geverifieerd ontwerp of model en 50% minder bugs dan bij rechtstreeks RTL-design.

HANS VAN THIEL

Het begon met een proefschrift van Xaio Wei Shen en James Hoe bij het befaamde MIT. De een is nu verbonden aan IBM Research en de ander aan de Carnegie Mellon universiteit, maar beiden werkten bij de MIT-groep van professor Arvind die in februari 2007 onderscheiden werd als ACM Fellow voor zijn werk aan dataflow-computing en verificatie. Arvind was in 1994 al benoemd tot IEEE Fellow en ontving tevens de Charles Babbage Outstanding Scientist Award. Hij richtte in 1998 het bedrijf Sandburst op, waar aan de initiële versie van de Bluespec-technologie werd gewerkt. Sandburst ontwikkelde 10G-chips en is nu onderdeel van Broadcomm. In 2003 werd het werk aan EDA-tools voortgezet in de nieuwe onderneming Bluespec.

In Bluespec opereert de ontwerper op 'electronic system level' niveau. Hij of zij specificeert het gedrag van het systeem en vervolgens kunnen de tools daaruit een ontwerp genereren op 'register transfer level' en/of een simulatie. Het bedrijf claimt naadloze integratie met ontwerp-flows van Cadence, Synopsys, Mentor en Magma, waaronder verificatie, debugging en synthese. De huidige Bluespec directeur Shiv Tas-



Figuur 1. Een mogelijke designflow met de Bluespec EDA tools. (Bron: Bluespec)

Bluespec-directeur Shiv Tasker: "Er is een verschil met andere FSM-modelleringen".

ker heeft, onder meer, negen jaar ervaring bij Cadence Design Systems.

Toestandsovergangen

De listing geeft een heel eenvoudig voorbeeld van een Bluespec-beschrijving, een 8-bits teller (uiteraard is de dimensie parametriseerbaar).
Een dergelijk programma wordt (onder meer) geëvalueerd door middel van termherschrijving, een formele methode die gebruik maakt van substituties van formules. Het model is gebaseerd op overgangen in een eindige toestandsautomaat (finite state machine) maar er is een verschil met andere FSM-modelleringen, zo licht Bluespec CEO Shiv Tasker toe. De ontwerper beschrijft geen unieke FSM, maar alleen toestandsovergangen met de condities waaronder die kunnen plaatsvinden. In termen van een toestandsdiagram: alleen de kanten worden beschreven, als transities ofwel regels. De crux hierbij is dat elke regel als op zichzelf staand wordt gecodeerd; de aanname is dat de rest van het systeem inactief (quiescent) is.
Uit de aggregatie van al die regels creëert de Bluespec-compiler nu een unieke FSM die aan alle beschreven condities voldoet en tevens zoveel mogelijk regels maximaliseert binnen een klokcyclus, zo vat Tasker samen.
De compiler-technologie gebruikt naast termherschrijving nog andere formele technieken, zoals BDD (Binary Decision Diagram) en eliminatie van gemeenschappelijke sub-expressies in verschillende stadia van checking en optimalisatie.

De Bluespec-tools zelf zijn geschreven in de functionele programmeertaal Haskell en de Haskell-compiler wordt gebruikt voor structuren en statische checking. Tevens krijgt de systeemontwerper met de Bluespec-tools een aantal Haskell-functies cadeau die niet beschikbaar zijn in SystemVerilog. Veel van die functies zijn wel beschikbaar in C++, maar C++ compilers kunnen geen type-checking leveren die vergelijkbaar is met die van de Haskell-compiler, aldus Shiv Tasker.

Tools

De achterliggende technologie bevindt zich echter onder de motorkap van de EDA-tools. De atomische transacties of regels kunnen worden geschreven in een versie van SystemVerilog of een C++ template dat Bluespec ook levert.
Als de C++ template class wordt gebruikt, dan wordt de code door de Bluespec ESEPro-scheduler omgezet naar een parallel programma dat door de Gnu C++ compiler kan worden gecompileerd.
Als alternatief kan de code door Bluespec ESEComp worden omgezet naar standaard Verilog 95 RTL-uitvoer. In de SystemVerilog syntaxis kan de beschrijving door Bluesim direct worden geëxecuteerd, of door de Bluespec-compiler worden omgezet naar Verilog 95 RTL.
De compilatie naar RTL wordt in twee fasen uitgevoerd. In de eerste fase wordt de oorspronkelijke ontwerpbeschrijving na uitgebreide statische typechecking omgezet naar een TRS (Term Rewrite System) dat kan worden ingeroosterd (scheduling).
In de tweede fase probeert de compiler zoveel mogelijk termen in dezelfde klokcyclus onder te brengen. Daarbij wordt gekeken naar resourceconflicten en wordt het ontwerp ingeroosterd in klokcyclussen om racecondities te vermijden.
Daarbij wordt automatisch muxing en control logica gegenereerd die niet in het oorspronkelijke ontwerp aanwezig was. Uiteraard is dit stuurbaar door de gebruiker, zo zegt Tasker, maar omdat die code wordt afgeleid of samengesteld door de compiler spreekt Bluespec van een synthesetool en niet van een codegenerator.

De SystemVerilog versie van een ontwerp kan direct worden gesimuleerd met de Bluespec Bluesim-simulator en de gesynthetiseerde Verilog 95 RTL is simuleerbaar met elke commerciële Verilog-simulator. Omdat er een vraag was van klanten naar virtuele platformen voor software-testing is in 2006 de technologie ook overgebracht naar C++/SystemC. Deze versie is simuleerbaar met elke SystemC-kernel, ofwel die van OSCI (Open System C Initiative) ofwel een commerciële simulator die OSCI SystemC ondersteunt.
Bluespec heeft kortgeleden een pakket ESL-synthese-uitbreidingen voor SystemC vrij beschikbaar gesteld, te downloaden van de Bluespec website.

Praktijk

Het bedrijf claimt een winst van 50% in ontwerptijd met 50% minder bugs dan bij rechtstreeks RTL-ontwerp. IP-ontwikkelaar Interra Systems heeft 25 ontwerpen, variërend van een verkeerslichtcontroller en een sequentiële vermenigvuldiger tot een random number generator, getest met de Bluespec-tools. De ontwerpen kwamen uit de designdatabase van het bedrijf en maakten deel uit van een suite voor het testen van EDA-tools. Een team van drie ontwerpers die geen eerdere ervaring hadden met de designs, ontwikkelde BSV (Bluespec System Verilog) versies en een BSV-testbank voor C-verificatie. Het kostte de engineers, die allen ervaring hadden met Verilog, drie weken om voldoende vertrouwd te raken met de BSV-taal en omgeving, en vier tot vijf weken om de 25 ontwerpen te coderen in BSV. Het ontwikkelen van de testbank vergde nog eens twee weken.
Enkele conclusies waren dat de code zeker gelijkwaardig is aan de bestaande, handmatig geschreven RTL-ontwerpen en dat de Bluespec-compiler zelfs onverwachte optimalisaties introduceerde. Het werd ook als een voordeel gezien dat met één ontwerp op hoog niveau zowel een RTL-model als een C-model beschikbaar komt. Tenslotte

levert ontwerpen met Bluespec een design-bibliotheek op die eenvoudig kan worden aangepast en die het onderhoud vergemakkelijkt.

Bluespec is een commerciële onderneming maar ook aan het onderzoeksinstituut van professor Arvind, waar het allemaal begon, wordt nog steeds aan de ontwerpmethodologie gewerkt. In 2005 is daar een MIPS I integer ISA-processor ontworpen met behulp van de Bluespec-tools. Het doel was om ook grotere designs effectief te kunnen beschrijven, debuggen en optimaliseren met deze EDA-technologie. ■

Referenties

Interra Systems, Inc., Bluespec Testing Results: Comparing RTL Tool Output to Hand-Designed RTL, 2004, beschikbaar op www.bluespec.com

Massachusetts Institute of Technology, Nirav Hemant Dave, "Designing a Processor in Bluespec", 2005, (csg.lcs.mit.edu/pubs/memos/Memo-484/memo-484.pdf)


```
interface Counter;
    method Bit#(8) read();
    method Action load(Bit#(8) newval);
    method Action increment();
endinterface

(* synthesize *)
module mkCounter(Counter);
    Reg#(Bit#(8)) value <- mkReg(0);

    method Bit#(8) read();
        return value;
    endmethod

    method Action load(Bit#(8) newval);
        value <= newval;
    endmethod

    method Action increment();
        value <= value + 1;
    endmethod
endmodule
```

Listing. Een Bluespec System Verilog beschrijving van een 8-bits counter. De bit-breedte kan uiteraard ook worden geparametriseerd, bijvoorbeeld in een size_t definitie. (Bron: BSV101: Designing a Counter, Bluespec 2005, beschikbaar op www.bluespec.com