

Tidbits

Python als hardwaretaal

6 april 2010

In 2003 begon Jan Decaluwe met het ontwikkelen van MyHDL, een manier om IC's te ontwerpen in een hoogniveauprogrammeertaal Python in plaats van de laagniveautalen VHDL of Verilog. De methode is regelmatig gedemonstreerd in FPGA's, maar nu ligt er een heuse commerciële Asic die grotendeels ontworpen is met MyHDL - voor zover bekend de eerste. Decaluwe vertelt over de voordelen, nadelen en gedachten van ontwerpen met een hoogniveautablet.

De eerste werkende chip ontworpen in de Python-programmeertaal is een feit. Jan Decaluwe heeft het digitale gedeelte van een mixed-signal Asic volledig geschreven in MyHDL, zijn Python-gebaseerde hardwareontwerptaal. 'Vooral om aan te tonen dat MyHDL gebruikt kan worden voor industriële toepassingen en niet alleen maar speelgoed is', legt Decaluwe uit.

De Asic, van mixed-signalontwerper ICSense, is een op maat gemaakte interface voor sensoren en zal in grote volumes van de band rollen. Het IC meet de kleine sensorsignalen met een 16 bit ADC en haalt daarbij een absolute microvoltgenauwkeurigheid. ICSense heeft het IC samen met buurman Easics ontworpen. Over de applicatie en de klant kunnen de ontwerperhuizen niks zeggen.

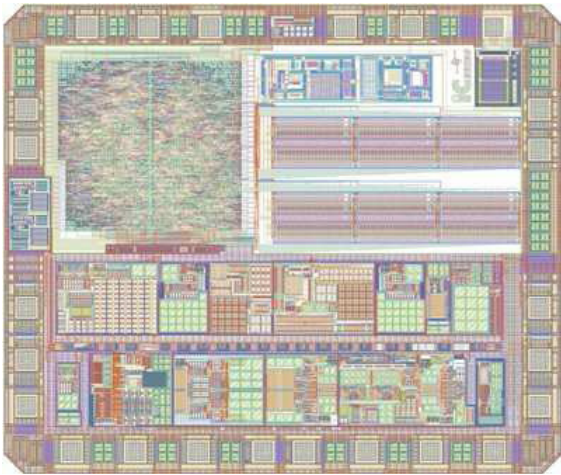
De Asic is een voorlopig hoogtepunt in de ontwikkeling van MyHDL. Decaluwe, medeoprichter van Easics, werkt sinds 2003 aan dit opensourceproject, een poging om de voortgang in softwareontwikkelingmethodologie van de afgelopen decennia naar de hardwarewereld te brengen. Naast een implementatie is het vooral ook een aanpak van hardware schrijven. 'In de software zijn belangrijke dingen gebeurd, bijvoorbeeld de opkomst van hoogniveauscriptingtalen zoals Perl, Python en TCL. Ik ben daar een enorme fan van, zeker van Python', vertelt de Vlaming. Ook zijn verschillende ontwikkelconcepten tot wasdom gekomen en hebben IDE's een grote vlucht genomen. 'Sinds de start van Easics in 1990 heb ik altijd het idee voor ogen gehad dat hardwareontwerp een soort van softwareontwikkeling is. Voor veel hardwareontwerpers is dat vloeken in de kerk.'

Hoogniveauprogrammeertalen worden zo genoemd omdat ze de koppeling met de hardware achterwege laten waarop de toepassingen draaien. Dat maakt features mogelijk waarmee de programmeur op een hoger niveau kan werken met zijn code. Waar het in C en C++ bijvoorbeeld nodig is expliciet geheugen te reserveren voor variabelen of databuffers, kan dat in talen als Python en Perl naar willekeur bijgeboekt worden. De programmeur hoeft zich daardoor veel minder met de geheugenhuishouding te bemoeien, wat resulteert in snellere ontwikkeling met minder bugs. De keerzijde is dat uitvoering van hoogniveautalen aanzienlijk trager is en meer geheugen vereist.

Randgevallen

MyHDL is een module in Python bedoeld om hardware te ontwerpen. Zo voegt het pakket klassen toe voor signalen of bitgeoriënteerde bewerkingen. Decaluwe ontwikkelt het pakket - op een enkele ingestuurde patch na - in zijn eentje, maar kan dankzij de keus voor Python meeliften op de snelstromende ontwikkeling van deze taal. 'Je moet een manier hebben om heel veel kleine stukjes hardware te beschrijven die parallel aan elkaar werken. Toen ik hiermee begon, werd in Python net het concept van generators geïntroduceerd. Dat zijn functies die hun toestand bewaren tussen elke aanroep. Dat komt eigenlijk perfect overeen met wat je nodig hebt in hardwaretalen', blikt Decaluwe terug op de ontwikkeling.

Ontwerpen in MyHDL zijn zowel naar Verilog als VHDL te vertalen. Het pakket ontsluit hoogniveauctechnieken voor hardwareontwerpers. Neem bijvoorbeeld het datatype voor integere getallen. De programmeur hoeft zich in MyHDL niet bezig te houden met de vraag of er 16 of 32 bits nodig zijn, of dat het getal negatief moet kunnen worden. Er is slechts één soort integer, dat tijdens de conversie naar het juiste HDL-type wordt omzet.



Een nieuwe mixed-signal Asic van ICSense en Easics zag het levenslicht doordat het digitale deel, linksboven op de chip, volledig ontworpen is in de hoogniveauprogrammeertaal Python. De ontwerper hoeft daardoor minder over implementatiedetails na te denken en kan abstracter omgaan met zijn ontwerp. Ook is met een druk op de knop naar believen Verilog of VHDL genereren, wat handig is voor integratie met bestaand IP. De grootste winst is echter dat de ontwerper dankzij MyHDL kan bouwen op de volledige Python-taal en -omgeving voor verificatie, validatie en testcases.

Toch is daar niet de grootste winst te halen. Hardware ontwerpen blijft hardware ontwerpen, dus de restricties gelden daar net zo hard voor in Python - alhoewel MyHDL geen extra beperkingen oplegt. 'Puur om RTL te schrijven, ga ik niet beweren dat MyHDL fundamenteel voordelen heeft ten opzichte van direct VHDL of Verilog. Maar voor dingen als krachtige testbenches, modelleren op hoog niveau en simulatie kan je volledig uitpakken. Daar heb je volledig Python tot je beschikking met de alle bibliotheken. De power die je dan hebt, is echt van een andere orde', vertelt Decaluwe.

De nieuwe chip is daar een sprekend voorbeeld van. Decaluwe gebruikte de testgedreven aanpak. Kort gezegd komt die erop neer dat de ontwikkelaar voor kleine stukjes software tests schrijft die automatisch kunnen worden uitgevoerd. Idealiter wordt er voor elke specificatie een testcase geschreven, nog voor de daadwerkelijke implementatie. Al die tests moeten slagen om de software van het stempel 'gereed' te voorzien. Deze methode is populair in softwareontwikkeling; Python heeft zelfs standaard een raamwerk aan boord om dit soort tests uit te voeren. Daar heeft Decaluwe dankbaar gebruik van gemaakt. Een voorbeeld hiervan is de decimatiefilter in de Asic, die het aantal samples per seconde omlaag brengt. Het gewenste gedrag is eenvoudig in Python te beschrijven, waarbij ook alle randgevallen zonder veel pijn in de test kunnen worden verwerkt. Gedreven door het testpakket kon Decaluwe de ingewikkeldere implementatie verzorgen.

Verhit

MyHDL staat daarmee vooraan in de ontwerpketen. Voor de laagniveausimulaties en uiteindelijke synthese is Verilog- of VHDL-code nodig. Wel bevat MyHDL

een hoogniveausimulator als Python-klasse om het hardwareontwerp programmatisch te testen. 'Er is één groot nadeel aan MyHDL, en dat is simulatieperformance. Python is niet goed om heel veel regressietests te draaien en zeker niet geschikt voor *gate level*-simulaties of heel nauwkeurig timing-simulaties', verklaart Decaluwe.

Bij deze laagniveausimulaties kwam een ander voordeel van de MyHDL-aanpak naar voren. Een eeuwig probleem in hardwareontwerp is dat zowel Verilog als VHDL als standaardtaal gelden. Verhitte debatten rond de merites van de een of de ander ten spijt zijn beide ongeveer even populair en in de praktijk worden ze vaak door elkaar gebruikt in een project, vooral dankzij IP van derde partijen. Zo ook bij het ontwerp van ICSense en Easics. 'Ik kon in Verilog of VHDL converteren naar gelang de nood. Easics is een VHDL-huis, maar ICSense wilde bijvoorbeeld ook een keer wat testen in Verilog. Toen kon ik dat gewoon in aanleveren', zegt Decaluwe.

Pieter Edelman

[Terug naar overzicht](#)

© Bits & Chips | Deze pagina op internet: <http://www.bits-chips.nl/nieuws/bekijk/artikel/python-als-hardwaretaal.html>