From silicon nanowire reconfigurable transistors to operating-system controlled circuit space.

Michael Raitza[a], Marcus Völp[a]

[a]*Technische Universität Dresden, Inst. for Systems Architecture, Nöthnitzerstraße 46, 01187, Dresden, Germany*

Reconfigurable hardware is one of the enabling technologies that may bring us application performance beyond the predicted end of Moore's Law. Future applications may carry descriptions of accelerators tailored to speed up their most critical operations, which can be loaded and run just as operating systems load applications and libraries today. System software may relocate functionality around damaged transistors by reconfiguring different parts of the chip to take over the required task. In this extended abstract, we sketch our envisaged roadmap from silicon nanowire reconfigurable field effect transistors (SiNW RFETs) all the way up to the operating-system functionality required to host and isolate applications of mutually distrusting users.

Their design simplicity give SiNW RFETs [1] strong advantages over standard CMOS circuitry: (1) they are in-place reconfigurable as p-gated and n-gated FETs using the same means that drive functional logic; (2) there is no forced electrical distinction between reconfiguration logic and functional logic; and (3) reconfiguration takes place at a sub-CMOS level with a speed in the order of switching the logic.

Because of (1) and (2), we expect to reuse as memory or other functional structures part of the circuitry that is traditionally reserved for driving the reconfiguration. To exploit this structural overlap between the non-functional (reconfiguration-only) and the functional hardware domains, we are currently developing a network simulator for SiNW RFET circuits to identify common patterns in basic CMOS circuits such as adders, memories and routers. Our goal is to show quantitatively where in the design space between no configuration and fully generic circuits (such as LUTs in FPGAs) RFET reconfiguration performs best.

Once these basic blocks are identified and layouts tried using traditional FPGAs as simulators until RFET structures become available, the next challenge is to safely and reliably grant not necessarily trusted applications access to the RFET configuration space. Should we restrict building blocks to inherently safe circuits or can we achieve better performance if we apply other means to prevent malicious or erroneous applications from installing circuits to spy or tamper with other circuits? Possibilities range from accepting only *configware* (i.e., descriptions of to be installed configurations) that are signed by trusted third parties to checking correctness proofs prior to reconfiguration that are carried with the description [2]. However, even if functional correctness and the absence of cross wire sensory is ensured by proof-carrying circuits, applications may install circuit building blocks that grant them access to the code and data of other applications. Besides managing circuit space as a novel resource, the primary task of an operating system for RFET-based reconfigurable systems is therefore to isolate applications of mutually distrusting users. We will investigate operating-system provided isolation building blocks such as intelligent DMA controllers to restrict which memory pages (and parts of the configuration space) applications may address. We present early results at the poster.

References:

[1] A. Heinzig, et al. "Reconfigurable silicon nanowire transistors", NANO Letters, 12 (Nov. 2011), 119—124.

[2] G. C. Necula "Proof-carrying code" ACM Symp. on Principles of Programming Lanuages (NY, USA, 1997) 106—119.