Xilinx SDNet:

A New Way to Specify Network Hardware

By Loring Wirbel
Senior Analyst

March 2014

The Linley Group

www.linleygroup.com

This paper examines Xilinx's SDNet specification environment and its role, both in defining elements in Software-Defined Networks, and in implementing reconfigurable network elements in both control plane and data plane.

Network Processing, Before and After SDN

The recent stampede to Software-Defined Networks defined by such protocols as OpenFlow, has roots as deep as the earliest network processors that emerged at the end of the 1990s. Segmentation of inline packet-processing duties from the control plane of the network node caused both network equipment OEMs and their semiconductor suppliers to rethink the way that tasks were processed by routers and switches. Early generations of communication processors that utilized on-chip control units were supplanted by devices that emphasized efficient packet forwarding and traffic management.

Control-plane processors, often based on RISC architectures such as MIPS or PowerPC cores, always emphasized programmability. As designers became more comfortable with optimizing packet forwarding tasks in the data plane, they identified many tasks that could be programmed and upgraded via firmware, such as header parsing and bit-field manipulation. A few traffic management and search table tasks were assigned to lookaside processors that tended to be implemented in ASSPs and fixed for size and function, such as the table size in search engines based on ternary CAMs.

Through the course of more than a decade of NPUs and soft network functions, Xilinx has worked on the SDNet Software Defined Specification Environment for Networking. Creation of domain-specific specification environments began in the middle of the last decade as a research project under Dr. Gordon Brebner. When the communication business unit was formed at Xilinx in 2010, SDNet was identified as one environment to commercialize in conjunction with the project to rewrite the Xilinx design environment, later released commercially as Vivado.

Because SDNet is agnostic to protocol, hardware implementation details, or performance scaling, its flexibility is unique in the industry. Many attempts have been made by academia, networking OEMs, NPU vendors, and EDA vendors to define packet description languages, parsing languages, and high-level tools to realize some of these capabilities in designing soft network elements. None, however, come close to matching the feature set of SDNet.

The Changing Nature of the Soft Network Element

Prior to the rise of OpenFlow, a router was a router and an Ethernet switch was an Ethernet switch. In the era of well-defined network elements, an ASIC often represented the most optimized hardware for a specific task of aggregation and traffic management of packets. Today, as OEMs take special pride in declaring their network elements to be soft, the hard-coded ASIC is vanishing rapidly in favor of a mix of FPGAs, network processors, and multicore processors. Software tools for programming such processors

must be diverse by their nature, because their roles are different for each type of chip and software vendor, and have shifted over time as the nature of programmability itself has changed.

In the middle of the last decade, some early research programs out of academia attempted to describe packets in object-oriented and markup-style syntaxes. Many of the projects, such as NetPDL and Packet Details Markup Language, sought only to abstract the behavior of protocols from Layers 1 to 7 in the OSI stack, and did not deal with instantiation of specifications in hardware.

Click Modular Router

One of the first true open specification standards came out of MIT, in a project called Click Modular Router in the middle of the last decade. The Click system is intended to aid in router configuration and to schedule threads where a router architecture has been defined, but it also aids in configuration parsing, giving it the potential for extensions into general packet parsing. Xilinx incorporated elements of Click as early as 2003, and published a paper on the resultant hardware at the 2004 DAC conference. The Click system was developed further by UCLA and by startup Mazu Networks, a company acquired by Riverbed in 2009, but has not been widely adopted in succeeding years.

SDNet and the Direct Connection to Hardware

Xilinx had ambitious goals for SDNet to be closely coupled with Vivado when the environment was under development. An initial intent to generate custom hardware components from a software description was enhanced by SDNet's ability to generate firmware for programmable elements within a packet processing block. It also had the ability to integrate third-party components. In all instances, the environment needed to perform these tasks without ties to a specific process technology or bus speed.

Xilinx developers wanted the specification software to be able to perform hitless updates, making changes to the underlying hardware between packets in a real-time packet transport, with no disruption to line-rate service. The software also had to be able to generate debugging and validation test benches.

In creating this high-level specification environment, Xilinx wanted to ensure that designers could input specific speeds, parameters, and behaviors the underlying hardware was to achieve, independently of the high-level functional description, even if these designers knew next to nothing about underlying FPGA architectures. In that sense, vertical-domain software specification environments developed alongside Vivado bring the power of Vivado tools to a design community more versed in vertical application fields such as communications and image processing, than in the minutiae of VHDL-level FPGA design. In creating a high-level specification environment as a front end for the silicon design flow, Xilinx has provided the system specialist familiar with software engineering methodologies the means to apply those skills to hardware optimization.

Prior to the launch of Vivado in 2012, Xilinx's integrated software environment (ISE) utilized tools, such as synthesis and place-and-route, acquired from a variety of startups. The intent in re-architecting the design environment was to provide a unified data model and a Tcl-based scripting language to each step in FPGA design. The unified data model allowed parts of an FPGA design to be captured simultaneously and cross-probed at different stages in the design process. Synthesis could provide RTL-level simulation models, or C-based algorithmic IP cores. Xilinx's recent adoption of the OpenCL language extended this ability to parallel multicore designs.

SDNet and OpenFlow

The re-architecting of Xilinx ISE into Vivado was taking place just as the first router and switch manufacturers interested in software-defined networking adopted an academic project which became OpenFlow. The intent in creation of the OpenFlow protocol was to separate the control-plane elements which directed the behavior of a packet switch from the forwarding plane hardware switch itself, usually locating the network-domain controller within a server related. One controller would have a universal view of all forwarding nodes, and would push flow-table entries to these nodes, which would take the specified action for a given flow. While the first OpenFlow version came from a joint team at UC Berkeley and Stanford University, its roots come from a 2006 project at Stanford called Ethane. The Open Networking Foundation was created to support OpenFlow in 2011, and Version 1.2 of OpenFlow was released under the foundation's auspices. While ONF founders were dominated by data center leaders such as Google, Facebook, and Microsoft, ONF members now include the likes of Cisco, Juniper, F5 Networks, Huawei, NEC, IBM, Dell, and HP.

For the data center manager, opening a forwarding-plane interface for a hardware switch allows the purchaser of networking equipment to no longer be at the behest of closed architectures. Router clusters and vertical hierarchies could be reconfigured on the fly, changing topologies and packet-forwarding rates as traffic characteristics changed.

For the user of a specification-driven design environment for networking equipment, the broad shift to OpenFlow and other SDN control protocols represented an ideal test case for SDNet. Hardware could be reconfigured up to the point of FPGA implementation, and even changed following instantiation in silicon through updates in firmware. SDNet's role in optimizing control plane and forwarding plane is only one aspect of its utility in networking, however.

Because SDNet can allow hitless updates in firmware for programmable elements, even as it optimizes the throughput, table size, and port configuration for fixed-function elements like TCAMs, particular network nodes can have functions upgraded with no network downtime. Xilinx sees a broader environment beyond traditional SDN, in which SDNet is used to add intelligence to the data path. Comparison of the two methods is shown in Figure 1.

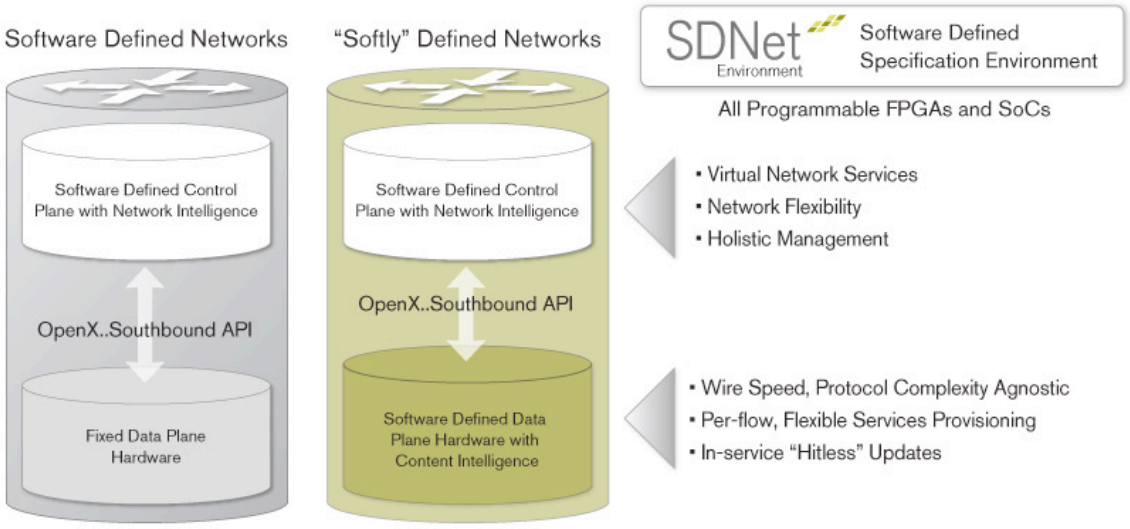

Figure 1: Broader provisioning tasks and hitless updates are possible with more content intelligence in a reconfigurable data plane.

The Competitive Landscape: Apples and Oranges

There should be little surprise that there are few tools seeking to perform the same network element configuration inherent in SDNet. The ASIC and FPGA communities, as well as general-purpose EDA companies, have done little research into tools for soft networking, and offer nothing comparable in tool suites. Vendors of NPUs and control-plane CPUs are anxious to provide tools that aid in programming through the use of higher-level languages, though they have no incentive for expanding the tools to allow for a level optimization in multicore utility that might improve overall throughput and thus lead to further erosion of NPU sockets. Similarly, some system OEMs in the router and switch worlds support OpenFlow to ease the operations of their own ASIC-based equipment, but do not expand into realms where the underlying hardware of their network elements might be modified or changed.

Merchant-Silicon Approaches

EZchip Technologies, provider of one of the highest-performance packet forwarding engines for more than a decade, is in the midst of a shift from its traditional data plane NP family to the new NPS, which turns to a multithreaded CPU running Linux and programmed in C. Traditionally, EZchip has treated its Microcode Development Toolset as an equivalent to similar control-plane CPU tools, with simulators, assemblers, and a library of subroutines. Other tools for frame generation and auto-generation of data structures are more common to datapath engines. The toolset for the future C-based NPS will likely look similar. Pre-characterization of EZchip architectures takes place through the use of application libraries, which simplify the programming of NP functional blocks

through predefined tasks such as Label Switched Routing, L2 Switching, VPLS, Firewall, and Access Control Lists. As such, the libraries are not that different from FPGA soft IP library elements.

Marvell offers two all-inclusive data plane software suites for its Xelerated processor. Both the Metro Ethernet Application and the Unified Fiber Access Application consist of an application package running on the NPU, and a control-plane API running on host CPU. The latter API is based on a hardware adaptation layer that includes boot scripts, configuration modules, and predefined messages to access the forwarding plane from control plane. This may represent an easy way for an OEM customer to bring a product to market, but modifications are only made through Marvell-supplied source code – the processor itself uses assembly code.

Cavium has opted for a more generic Software Development Kit for its Octeon family. The SDK includes a Gnu toolchain, simulator, Cavium's own ViewZilla for graphical analysis, and a regular-expression pattern compiler for deep packet analysis. Vertical software toolkits provide C-based routines for SSL, TCP, IPsec, and similar common Layer 3-5 functions. A general-purpose control plane CPU with some datapath elements can be highly flexible, but is far from optimal for packet processing.

Freescale has the most years of experience with its PowerQUICC and QorIQ families, and offers a wealth of software tools to optimize the mix of control-plane and datapath engines offered. However, the Processor Expert, CodeWarrior, and VortiQa families can represent a daunting tool suite, which do not offer full integration with each other, and are not designed to support all SDN concepts. Each MPU and MCU family in Freescale has a unique Processor Expert software suite, with the most relevant being the QorIQ Optimization Suite to configure the multicore processor for a specific task. It can be linked to CodeWarrior Development Studio for specific CPU architectures, a studio that represents a comprehensive coding platform for control-plane tasks. The VortiQa suites for building vertical applications have moved closest to SDN, as Freescale has launched a specific VortiQa for SDN, which includes Open Network director software and Open Network switch software. The SDN application suite for VortiQa, as with other Freescale suites for SMB, wireless infrastructure, and the like, are useful for characterizing existing communication processor architectures. Freescale's Data Path Acceleration Architecture offers many configurable elements in different members of the processor families, and VortiQa tools help to increase the utility of coprocessors. But this is not an all-soft compilation solution similar to what Xilinx offers with SDnet.

Network Equipment OEMs

Manufacturers of switches and routers always have treated SDN with a certain amount of trepidation, as it commoditizes the very network elements that represent the heart of their business. Nevertheless, architectural leaders like Cisco and Juniper have recognized the inevitability of SDN, and are opening more programming and configuration tools to their customers. In some cases they may help customers decide how to allocate packet-forwarding cards and I/O line cards across a midplane or backplane, but allowing a

customer to directly decide on hardware reconfiguration in real time may be expecting too much altruism from an OEM.

Cisco, for example, has enhanced its configuration software suites for switches and routers with an eXtensible Network Controller (XNC) program that eases a transition to SDN by keeping a traditional control plane and adding an external controller, with a hybrid software package called Monitor Manager. At the same time, its most complex ASICs for core routing are offering more hooks into user configurability. As the company upgraded from QuantumFlow to nPower X1 for its newest core routers, Cisco promised in-field hitless software upgrades with no packet loss, and better customer data path control for implementing full OpenFlow SDN. In all cases, however, this refers to more customer ability to program ASICs through firmware. Full OpenFlow support in hardware is not yet offered by Cisco.

Juniper has added more customer configuration flexibility since deploying its Trio chipset four years ago. Separate data path processors for lookup, memory, and queuing can be repurposed to a certain extent in MX routers – but not changed in basic hardware capabilities. In late 2013, Juniper introduced the Contrail SDN controller, based on its acquisition of Contrail Systems, which allows the virtualization of enterprise data center or service provider physical network resources. A new service provider package, Junos Fusion, allows some additional resource management, but only above the granularity of a particular router or switch.

In the first few weeks of 2014, competitors followed suit, including Huawei with its SoftCOM virtualization tools, and Alcatel Lucent with Cloudband, both based on expanded visibility of open nodes such as Session Border Controllers. But in all cases, SDN-based reconfiguration refers to network topology, not underlying hardware. This only makes sense, as OEMs would want to keep the open interfaces between control and forwarding planes at as granular a level as would be acceptable to service providers and data center managers, in order to protect existing platform technologies.

EDA Tools and ASIC Vendors

As structured ASICs crafted by dedicated design tools have given way to a 21st-century ASIC industry bearing more resemblance to a pure foundry business, commercial tools from EDA vendors are the primary option for OEMs continuing to work with non-FPGA customized silicon. Consequently, the only potential design methodology for a networking OEM working with ASICs comes from the EDA industry. Developers at Cadence, Mentor Graphics, Synopsys, and smaller competitors seem unaware or uninterested in SDN trends.

Synopsys's DesignWare for System on a Chip design comes closest to meeting soft networking goals, albeit more by accident than intent. The company only recently added 40G Ethernet controller IP to its DesignWare for Data Center SoC packages, and has not discussed the virtualization of network functions using SDN. Synopsys and Cadence try to offer generic Ethernet and OTN IP, often through third-party IP suppliers, while

Mentor has focused more on system-level design that has moved away from networking in recent years. No EDA company has made SDN a critical goal for upcoming releases.

This may reflect the uncertain role ASICs or EDA-driven custom silicon will play in network equipment of the future. The major networking OEMs develop their own NPUs with custom data paths, while startups are shifting from ASICs to FPGAs and merchant NPUs.

FPGA Design Tools

Altera and Tabula represent the two Xilinx competitors with large complex devices that target the networking space, though the design suites from the companies show little intent to move to SDN support. Altera has strengthened the ARM portions of its Quartus II design environment by adding the development kit for OpenCL, a high-level means of achieving parallelism. Both Xilinx and Altera utilize OpenCL. It is important to remember that OpenCL is general-purpose and not customized specifically for networking data flows. To date, individual tools within Altera's Quartus II, such as Qsys and DSP Builder, aid in the compilation of pieces of a design, albeit without a driving goal of in-field hitless updates of the network.

Tabula's Stylus compiler provides synthesis and placement to combine different logic planes to the unique multiplexed architecture of ABAX2 the company calls "Spacetime," but the unusual FPGA architecture is not easy for a neophyte to program. Last year, Tabula released Ethernet parsing evaluation kits to increase interest in ABAX2 for networking. New soft IP includes a scheduler and a 400G bridge, but all are based on fixed L2 Ethernet functions designed with Stylus.

Xilinx SDnet as a Hint of Networking's Future

The SDNet environment gives a preview of the approach Xilinx plans for other vertical domains. Taken as a whole, the Xilinx SD* environments could radically change the meaning of programmability, bringing back the notion of a reconfigurable architecture that can be changed on the fly. The capabilities of SDNet arose just as the OpenFlow model was finding favor, but the implications of SDNet outside the OpenFlow SDN world, and even outside Xilinx 7 and UltraScale architectures, could be significant. SDNet could become a model for how hardware is re-architected for different tasks, both in firmware reprogramming and in changing the configuration of IP cores. A typical flow from SDNet specifications and compilation to instantiation in a line card is shown in Figure 2.

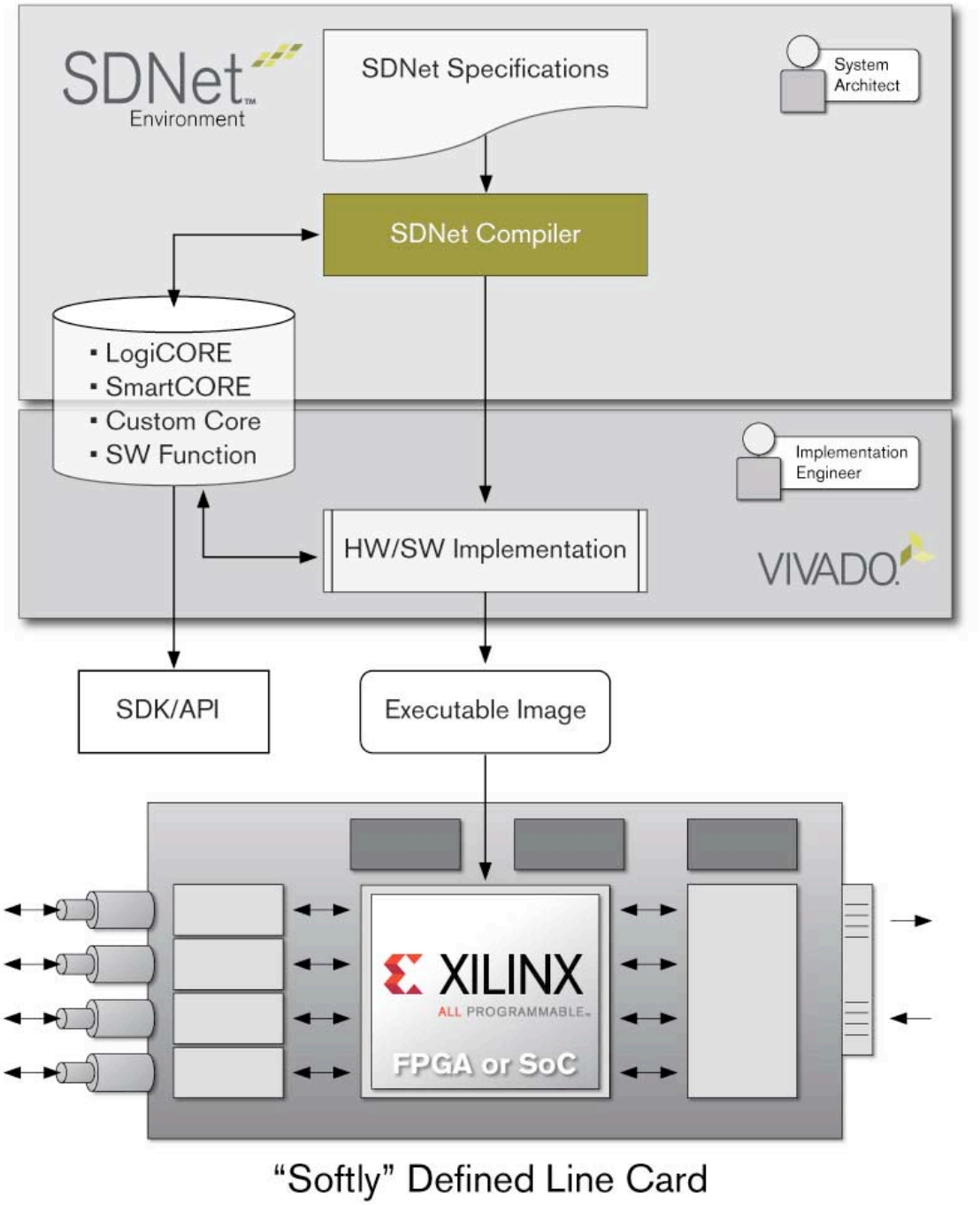

Figure 2: The executable image created through SDNet compilation can be implemented in Xilinx FPGA or SoC.

SDNet's capability of interconnecting hierarchies of specialized engines, such as QoS policy engines and search engines, provides a model for how those versed only in high-level syntax can create a dynamically reprogrammable architecture with linked data plane engines and coprocessors. The design flow Xilinx utilizes, from functional specification to compilation to VHDL, certainly is a common one in the industry. It would be likely to see other developers attempt to mimic Xilinx's specification methodology. The company has hinted it might open elements of SDNet to standards bodies, in a format dissociated from specific Xilinx 7 and UltraScale FPGA instantiations.

For the time being, SDNet has no direct competition because SDNet represents a radical break in implementing SDN capabilities in hardware that can be reconfigured for a particular task at hand. Certainly it will be useful in OpenFlow-based networks, but Xilinx has defined a tool with utility well outside the realm of what is typically considered SDN. ♦

About the Author

Loring Wirbel is a senior analyst at The Linley Group. The Linley Group offers the most comprehensive analysis of the networking-silicon industry. We analyze not only the business strategy but also the technology inside all the announced products. Our weekly publications and in-depth reports cover topics including Ethernet chips, network processors, multicore embedded processors, and wireless base-station processors. For more information, see our web site at www.linleygroup.com.

This paper is sponsored by Xilinx, but all opinions and analysis are those of the author.