RAZOR: ADVANCED ARCHITECTURE FOR THUMB-SIZED SOFTWARE-DEFINABLE RADIO

Clark Pope (DRS Signal Solutions, Gaithersburg, Md; cpope@DRS-DS.com)

ABSTRACT

This paper introduces a novel software-definable radio (SDR) architecture, called Razor™. The Razor is a thumb-sized SDR composed of a commercial off-the-shelf (COTS) processing module mated to a low-cost, receiver based on a chain composed of a radio frequency (RF) front end and an application-specific integrated circuit (ASIC). The processing module contains a digital signal processor (DSP), 512 MB of random access memory (RAM), a secure digital card, a field-programmable gate array (FPGA) with 1 million gates, and a universal serial bus (USB) interface. The receiver chain consists of a ceramic preselecting filter, a single-chip upconverter and local oscillator (LO), a surface acoustic wave (SAW) filter, and an integrated downconverter and analog-to-digital converter (ADC). It can be configured to cover 20 to 2400 MHz with bandwidths up to 40 MHz and performance greatly superior to conventional direct conversion implementations. The device looks like a standard USB cellular modem, but houses a complete GNU Radio-based SDR system. The device can plug into any personal computer's USB port or operate on a stand-alone basis. When operated stand-alone most USB devices can be connected directly (via hard drive, Ethernet dongle, Wi-Fi®, etc.) or via a multiport USB hub. By putting the optimal superheterodyne receiver architecture in a USB device form factor, the Razor provides a unique mix of price; performance; and low size, weight, and power (SWaP). This paper also provides a brief history of SDR approaches and technology as well as a detailed description of the design objectives and choices made in developing the Razor architecture. The developmental toolset and environment are described. The device's intended use in low-cost, commercial/academic markets for electronic security, wireless sensor, unmanned aerial vehicle, and test and measurement applications are discussed.

HISTORY OF SDR TECHNOLOGY

DRS Signal Solutions, Inc. (DRS-SS) has been designing radios for nearly 50 years and SDRs for nearly 20 years. The state of the art in tactical SDR is called Picoceptor, whose architecture and development history is well documented [1].

While very successful at building high-end, high-performance, low-SWaP radios, DRS-SS has never focused on devices for the academic research and consumer/hobbyist markets. Recent advancements have prompted DRS-SS to enter these consumer markets. First, readily available signal processing hardware like that provided by Gumstix®[2] and Ettus Research[3] provide significant processing capacity at low cost. Open source SDR software like GNU Radio[4] and OSSIE[5] have matured. RF integration has advanced to the point that decent performing radios can be built with a handful of commodity parts as companies such as RF Micro Devices, Analog Devices, Maxim Integrated Products, et. al., increasingly improve their designs to approach the physical and geometric limits of what is possible. Finally, there is a proliferation of applications that only require moderate RF performance in mild RF environments related to production line test equipment, remote sensing, and academic research. This makes specifications like preselection, local oscillator (LO) radiation, and dynamic range less critical.

All of these market and technology developments have prompted the Razor architecture --a low-cost, moderate-performance, consumer-grade radio based on open source software and COTS digital and RF modules.

1. SDR REQUIREMENTS

A generic SDR system is shown in Figure 1. A detailed description of the generic system is documented[1]. As long as digital intermediate frequency (IF) data from the tuner, baseband in-phase and quadrature (I/Q) data from a digital downconverter (DDC), and demodulator data is available at the processing/control element, virtually any SDR application can be supported.

In addition to the generic architecture requirements, the Razor has very specific cost and performance targets that essentially impact all aspects of the design. First, the device should have excellent sensitivity, minimal LO leakage, decent dynamic range, moderately low phase noise, and good IF/Image rejection of 60 to 70 dB. This performance requirement mandates a conventional superheterodyne architecture as described in Section 2.1. To be useful the device must have a reasonably wide tuning range of a gigahertz and, more particularly, cover VHF/UHF bands where most ham and hand-held radio traffic resides.

Beyond performance, several requirements for Razor are driven by cost. Wall supplies and batteries take up space and cost money so instead the device is powered by a PC USB port. Software development is very expensive and costs tremendous amounts of non-recurring engineering to build and maintain. A low-cost device cannot support this kind of overhead so the device must operate using freely available open source software. Because digital technology and devices move so rapidly compared to RF/analog technology, it is important that the devices main processing element consist of COTS modules. It is simply cost-prohibitive to create and maintain a custom board design of any complexity for such a low-volume, low-cost market. To be manufacturable at low cost the design must not go beyond basic rapid prototype and assembly manufacturing capabilities. This generally means the device must use conventional printed circuit boards (PCBs) consisting of fiberglass reinforced epoxy laminates of no more than six layers, and use only components that are easily kitted (from Dig-Key, for example). RF shielding, aluminum chassis, and gasketing are very expensive; thus, the device must provide reasonable spurious performance in just a simple aluminized plastic housing. Finally, in order to reach a large enough market and to minimize regulatory costs the device must be designed specifically for international export.

The remaining requirements drive the device's usability. First, the device should be fully configurable; specifically, the FPGA, operating system, and application software should all be extendable and upgradeable with minimal effort. The device should have minimal SWaP to maximize the range of applications with which it can be used. Similarly, the ability to operate stand-alone, i.e. without an attached PC, is critical to expanding the range of applications and, therefore, the potential volume of users.

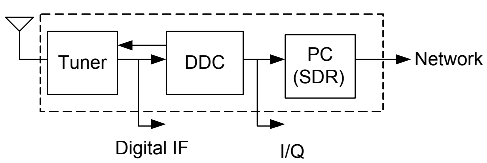

Figure 1. Generic SDR Platform

2. RAZOR ARCHITECTURE

This section provides some background on the technologies available and the ultimate design choices made for the Razor RF front end, signal processing hardware, and software.

The overall Razor architecture is shown in Figure 2. The RF input is put through preselection filtering and amplification to feed an integrated 1st LO and mixer. The IF SAW filter limits the bandwidth for selectivity. The Linear Technology LTM9005 [6] is an integrated module that contains a down converting mixer, IF SAW filter, and ADC. The ADC clock and 2nd LO are provided by fixed crystal oscillators. The Xilinx Spartan-6® FPGA processes and down converts the ADC to baseband digital in-phase and quadrature-phase (I/Q) data. Finally, the Gumstix Overo® Tide board hosts the Linux® operating system and GNU Radio application software for signal processing. The power and network communications is all carried over the USB plug.

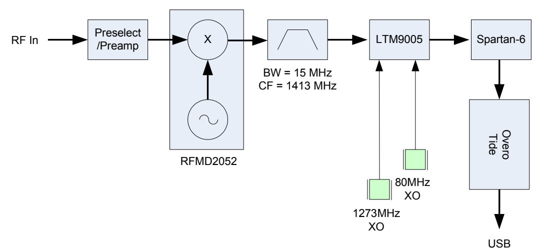

Figure 2. Razor Architecture

A mock up of the Razor device is shown in Figure 3.

Figure 3. Razor Mock Up.

The costed preliminary bill of materials (BOM) is around $300 if built in quantity. This includes the $169 Overo Tide module. It is believed that the device can be manufactured and sold profitably at three to four times the BOM cost. Also, the power consumption is estimated at 3-4 watts. This technically violates the USB port power specification of 2.5 watts; however, it is possible to reduce the clock of the Gumstix module. Most modern computers provide more power than the minimum called for in the USB specification, and two USB ports can always be used if additional power is necessary.

2.1. RF Front End

In general, three technologies currently exist to allow converting RF energy to baseband for processing: direct digitization, direct conversion using RF ASICs, and the traditional discrete superheterodyne receiver.

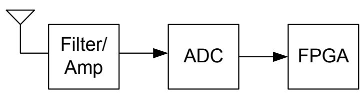

Figure 4. Direct Digitization

Direct digitization, shown in Figure 4, features low cost and relatively high dynamic range, but the upper usable frequency is limited as the jitter on the ADC clock degrades performance with increasing amounts of subsampling. In addition lots of preselection, either tracking filters or band select filters, is required to keep aliased frequencies out of the Nyquist bandwidth of the device.

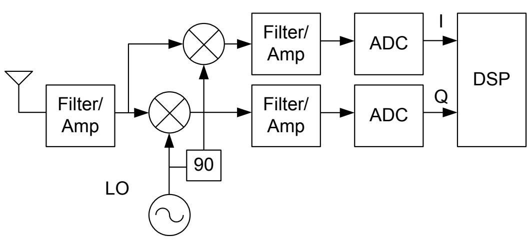

Figure 5. Direct Conversion.

Direct conversion is the method used in almost all RF ASIC based radios because it only requires one LO and one heterodyne mixer. See Figure 5. Though cheap and easy to integrate, this architecture suffers from several problems. First, because the LO is at the same frequency as the received signal and because the reverse isolation of the first mixer is limited, the radio radiates a portion of the LO out the antenna. This can interfere with both the desired signal and other users. Second, because it is extraordinarily difficult to phase and frequency match the distinct I/Q signal paths even with elaborate compensation, the spur-free dynamic range at the DSP is limited to 50 or 60 dB. Without preselection the entire RF spectrum is allowed to hit the input of the radio and can cause all manner of spurs and distortion as everything on air mixes with everything else. This is alleviated with a nice preselection filter but that limits the tunable range and increases complexity. Finally, the LOs that are common in the direct conversion devices usually have higher phase noise and more spurious than discrete LO designs.

In all, direct conversion can be cheap and effective with high sensitivity but generally only over relatively small chunks of the RF spectrum. For better performance over a wider range of RF frequencies the classical superheterodyne tuner, shown in Figure 6, is used in the Razor.

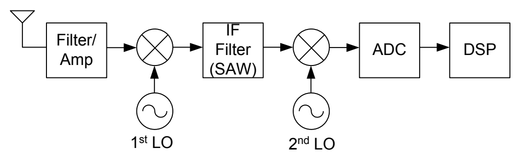

Figure 6. Superheterodyne Tuner

The superheterodyne tuner employs a 1^{st} LO and mixer to convert the RF input to a fixed intermediate frequency and then a 2^{nd} LO and mixer to convert back down to a frequency suitable for digitization. The advantages are three-fold. First, the IF filters can be designed for high selectivity and maximum image rejection. Second, neither LO is at a frequency that is easily conducted out the antenna. Finally, since there is no I/Q matching required, the spur-free dynamic range is limited only by the ADC device, which has typically 70 to 80 dB of range or more.

The drawback of the superheterodyne receiver is that RF signals at frequencies one 1^{st} LO frequency above the desired signal also mix to the IF frequency. Similarly, RF signals at the IF frequency can pass through the first mixer. These image and IF frequencies have to be filtered out prior to the 1^{st} mixer with some preselection.

The actual Razor tuner is a superheterodyne with some specific optimizations for cost and size as shown in Figure 2. The current design has an input frequency range of 20 to 980 MHz and an IF frequency of 1413 MHz This means that both the IF and the minimum image frequency at 1846 MHz are well outside the preselector filter. For low cost, two ceramic filters costing one dollar are used in cascade to provide the preselection. The first LO and mixer is an integrated device called the RFMD2052[7]. This devices has good linearity and decent phase noise and the LO frequency range can support RF inputs as high as 2400 MHz Two low-cost cascaded SAW filters provide the IF selectivity. The LTM9005 RF module from Linear Technology [6] performs the final

Table 1. Razor Radio Gain, Noise Figure and Intercept Point Analysis

No.	Razor Radio With LMT9005 Stage Description	Stage					Cumulative						
		Gain (dB)	NF (dB)	IIP3 (dBm)	P1 (dBm)	IIP2 (dBm)	Gain (dB)	NF (dB)	IIP3 (dBm)	IIP-NP (dBm)	P1 (dBm)	IIP2 (dBm)
1	Input Protection	-0.5	0.5	99.0	99	99	-0.5	0.5	99.0	98.5		
2	LPF LFCN-1000	-0.9	0.9	99.0	99	99	-1.4	1.4	96.2	94.8		
3	BLANK	0.0	0.0	99.0	99	99	-1.4	1.4	94.8	93.4		
4	PreAmp MGA82563	13.0	2.2	18.0	99	99	11.6	3.6	19.4	15.8		
5	LPF LFCN-1000	-0.9	0.9	99.0	99	99	10.7	3.6	19.4	15.8		
6	Active Mixer RF2052	-2.0	12.0	18.0	99	99	8.7	5.5	7.0	1.5		
7	Diplexer	-0.5	0.5	99.0	99	99	8.2	5.5	7.0	1.5		
8	1st IF SAW Filter TFS 1413	-2.7	2.7	30.0	99	99	5.5	5.7	6.9	1.2		
9	1st IF Amp	13.0	2.2	18.0	99	99	18.5	5.9	5.8	-0.1		
10	1st IF SAW Filter TFS 1413	-1.0	1.0	40.0	99	99	17.5	5.9	6.7	-0.2		
11	2nd Mix/IF/ADC LMT9005	0.0	16.0	17.0	99	99	**17.5**	**6.6**	**-1.4**	**-8.1**		

downconversion and sampling. To save space both the 2nd LO and ADC clock are oscillators from Silicon Labs, who are pioneering a quick-turn, low-cost, custom oscillator product. The oscillators are merely 3 by 5 millimeter, with great phase noise, and no problem being obtained at the custom frequency required for the Razor.

The gain, noise figure, and intercept point analysis of the tuner is shown in Table 1. Overall the gain and noise figure are good and the intermodulation linearity is acceptable.

2.2. FPGA

The selection criteria for the FPGA were low cost and power, while still having enough resources to host the processing in the existing Ettus universal software radio peripheral (USRP) FPGAs and to allow for additional expansion for some custom processing. The Xilinx Spartan 6 XC6SLX16-2FTG256C is available for only $24 in low quantities and has the same pinout as both the LX9 and LX25. It has 2278 slices, 32-block RAM, and 32 multipliers. Also, it has only a 17mm by 17mm footprint. An additional benefit of this device is that it is supported by the free Xilinx ISE Webpack® so users will not have to purchase the Xilinx tools in order to modify or recompile the FPGA.

The purpose of the FPGA is to host the standard processing in the Ettus USRP FPGA plus some additional down conversion that converts the real ADC data from the LTM9005 to baseband I/Q data that GNU Radio expects. Though a custom FPGA could be built from scratch, it is desirable to just insert a real to complex module before the standard U2_core.v Verilog module as shown in Figure 7. The USRP traditionally samples an analog I/Q waveform provided by a daughterboard with a dual ADC. By bypassing those inputs and substituting the single Razor ADC data path with real to complex conversion, the effects on the source code and upstream GNU Radio processing are minimized.

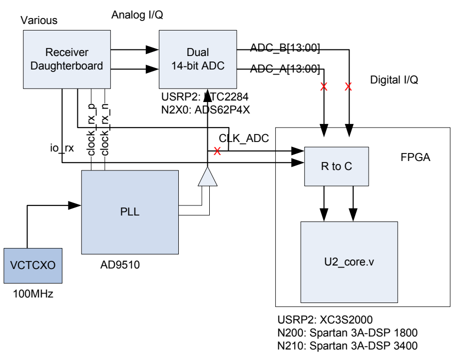

Figure 7. USRP FPGA Modification.

2.3. Processing Module

The selection of a processing module was very simple since there is only one module currently available known to support a standard GNU Radio distribution. That module is the Gumstix Overo Tide [2]. It is composed of a 720 MHz OMAP 3530 ARM/C64x+ processor, 512 megabyte RAM and microSD card slot. It contains its own power management and brings out signals for audio, external busses, USB, I2C, SPI, UART, and other interfaces to two 70-pin headers.

Although, technically, any member of the Overo family could be used, the Tide has the maximum available RAM memory, a plus for signal processing. It lacks support of the air interfaces (Wi-Fi and Bluetooth®) provided by other family members, so its power consumption and price are less.

2.4. Software

The Gumstix Overo modules use OpenEmbedded[8] to build the Angström distribution of Linux kernel, device drivers, root file system, and u-boot bootloader. The build relies on the bitbake tool to pull source code from all over the internet and to configure, compile, and install it in the final image files. Users can build custom images or just use the stock images provided by Gumstix.

If the omap3-desktop-image is built, the system will have a complete windowed environment as shown in Figures 8 and 9. The system has a package manager, called opkg, which can install thousands of prebuilt applications just like typical desktop Linux distributions. All the standard networking tools are installed by default: ssh, sftp, httpd, xvnc11, etc.

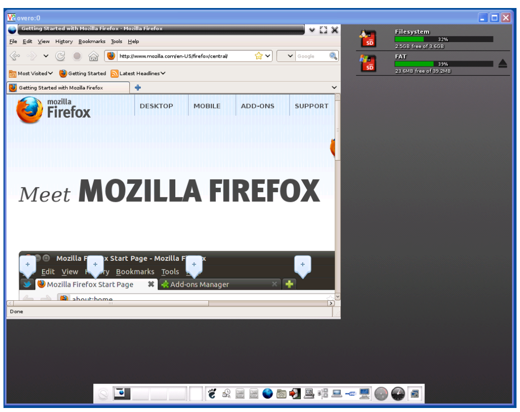

Figure 8. Gumstix Window Environment.

One of the prebuilt packages is GNU Radio , which also includes examples, drivers, and the GNU Radio companion (GRC). Mostly the GNU Radio installation is the same as for desktops. However, to get data from the FPGA into the GNU Radio framework a custom device driver is required.

A low-level custom device driver is provided that allows the user to control and access the Razor™ FPGA hardware. The result of this effort is a kernel module that contains functions that can be called by GNU Radio using standard Python programming. The device driver allows the user to change parameters such as frequency and attenuation that are mapped to FPGA registers.

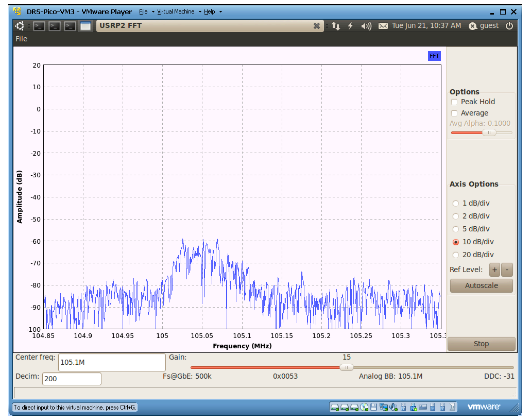

Figure 9. Gumstix Window Environment with GNU Radio Demo.

Linux also provides the user with the capability to write their own applications and load them onto the Razor. Users can place their own applications on the micro SD card on the Razor using secure file transfer protocol (SFTP) and execute them via a SSH session or run them as a Linux service.

Note that users are not limited to the Angstrom distribution or GNU Radio. Any operating system that runs on Gumstix and any SDR tool can be loaded onto the device.

2.4. Developmental Tools

The entire Razor developmental environment is contained in an Ubuntu 10.04 LTS virtual machine (VM) as shown in Figure 10. The LTS stands for long-term support meaning that this particular distribution will continue to receive updates and bug fixes for several years. This allows for minimal installation and ensures all Razor users are using the same OS and tools. The VM contains three virtual disks: a 10 Gbyte disk for the OS, a 20 Gbyte disk for the Xilinx Webpack tools, and a 100 Gbyte disk for the OpenEmbedded (OE) build system. The OE has to be very large since the build of a desktop Gumstix image can take as much as 65 Gbytes of disk, 2.5 Gbyte RAM, and take nearly a day to run. Because of the size of the VM, it can only be distributed practically via portable eSATA hard drives.

Since the Razor is powered and controlled via USB there are no special tools required; however, pads are provided to attach a 1.8V USB to serial cable in order to gain access to the Linux console and it is possible to attach a Xilinx JTAG cable for running Chipscope, for example.

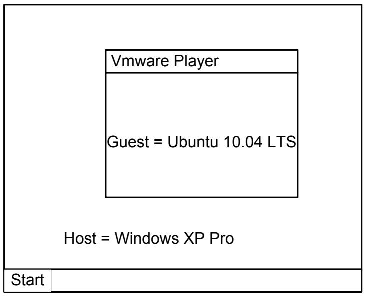

Figure 10. PDP VMware Image

It is important to note that Razor users are not limited to the given OE toolflow. Any OS that runs on Gumstix could be used. So far these include Windows CE® and Ubuntu®. Similarly, alternate FPGAs and device drivers can be designed to replace or augment the base functionality. The limits are based mostly on the developer's experience and existing IP.

2.5. CAD Tools

As mentioned in the Requirements section complying with common commercial design rule checks when designing the board is key to ensuring that the PCB is manufacturable at low cost. One way to guarantee this is to use a quick turn PCB house's own layout software, such as PCB123[9] from Sunstone. Figure 11 shows a 3D model of the bare PCB within the PCB123 tool.

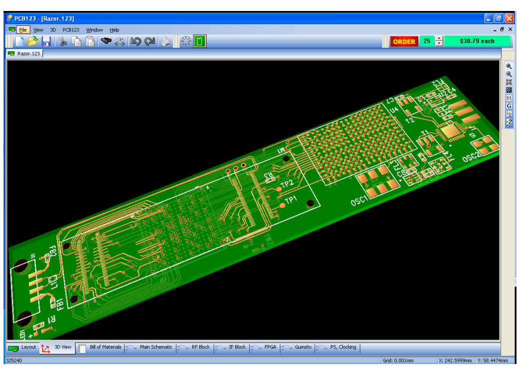

Figure 11. PCB123 3D Designer Viewer

Additionally, for the mechanical housing, 3D printing has become commonplace with several vendors providing online design and ordering software. An on-line service eMachineShop®[10] was used to prototype the housing for Razor. The housing was designed using their free CAD software, sent to the machine shop online, and the finished parts were received a week later.

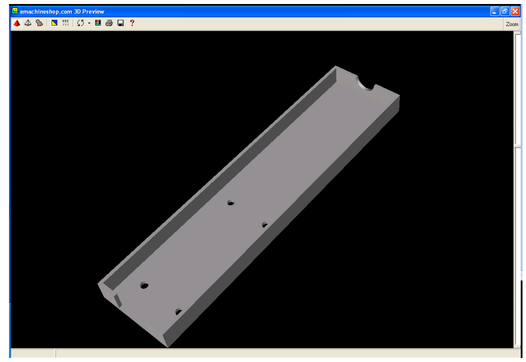

Figure 12. eMachineShop CAD 3D Designer Viewer

3. APPLICATIONS

With its moderate RF performance and its lack of rugged packaging or design for mitigating extreme environmental conditions, the Razor is only suitable for commercial and consumer applications. The applications are primarily academic research, production line testing, ham radio, and the like. The section describes several practical uses for the Razor.

The most obvious application for the Razor is product line testing for factories or depot repair shops. Here the manufacturers of cordless phones, family radio service (FRS) radios, Wi-Fi cards, etc. need a simple device to test the transmitters of their product. Since the test would run by cabling directly to the Razor and might contain a very good inline preselector filter, there is no danger of interference and the test does not require exceptional RF performance. Example tests would be power level and error vector measurements of a digital transmitter. These are easily scripted with GNU Radio.

Expanding on the test capability, the Razor could be programmed to become a generic spectrum analyzer. Because of the spurious performance of the device, some calibration routine would be required to find, characterize, and factor out the radio's own spurs. At a minimum the Razor could perform good differential RF measurements especially when coupled with a planned USB RF signal generator device. The Razor is even small enough that it could be integrated into a handheld configuration to create a portable spectrum analyzer such as one commonly used for electromagnetic compatibility (EMC) compliance testing. Note that the tune time is relatively slow at 5 milliseconds. This puts the maximum scan time at about 3 GHz/sec assuming a 15 MHz IF bandwidth.

The Razor, of course, can be useful in academic research into digital communications, cognitive radio, and digital signal processing. The device is cheap enough for most students to afford. Professors can use the Razor to sniff and record live radio communications to generate sample files for students to work on processing in lab assignments.

Finally, a battery and USB devices can be attached to create a stand-alone sensor node as shown in Figure 13. Since the Gumstix USB can operate as a host, virtually any USB device can be attached including WiFi dongles, hard drives, display units, single board computers, etc. This creates a very powerful sensor platform that is cheap enough to be used in leave-behind applications.

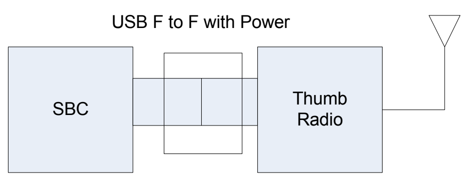

Figure 13. Stand-alone Configuration.

4. FUTURE WORK

At the time of writing, actual RF performance measurements and signal processing benchmarks were not available although both the RF front end and processing back end have been prototyped. Contact the author for updated performance data.

The Razor design required design choices to ensure economic viability that prevent the inclusion of an external reference which would be required to support multiple input/multiple output (MIMO) or phase-coherent collection. Additionally, the IF bandwidth is modest to keep the ADC clock rate low to save power. External synchronization and wider bandwidths are expected to be supported in future revisions of the design.

The Razor is designed to be modular so that alternate receivers and transmitters can be plugged into it. It is expected that direct conversion, HF, and superheterodyne with alternate preselection receivers will be developed. The architecture also supports the entire Gumstix Overo series of boards so that the processing can be upgraded as the Gumstix boards are upgraded. A board that replaces the Gumstix with just a gigabit Ethernet or USB 3.0 interface is possible. The base functionality is easily extended with custom FPGA designs and the basic hardware is easily migrated to newer/faster FPGA devices.

5. CONCLUSION

This paper has described the design objectives and considerations that went into the development of the Razor Thumb-Sized SDR platform. The architecture and development tools were discussed and several real-world applications were implemented and described for the platform. The Razor represents a novel mix of performance and cost for SDR technology. The architecture is adaptable and extendible and relies heavily on open source software, commodity processing hardware, and highly integrated commercial RF modules. Because of the Razor architecture's modular design, it is expected to continue benefitting from advancements in all areas of FPGA, SDR, Gumstix, and RF IC technologies over time.

6. REFERENCES

[1] Clark Pope and Mike Kessler, “Picoceptor: Advanced Architecture for Miniature Software Definable Radio Systems”, SDR Forum Conference, November 2008.
[2] http://www.gumstix.com/
[3] http://www.ettus.com/
[4] http://www.gnu.org/software/GNU Radio/
[5] http://ossie.wireless.vt.edu/
[6] http://www.linear.com/product/LTM9005
[7] http://rfmd.com/products/IntSynthMixer/default.aspx
[8] http://www.openembedded.org/index.php/Main_Page
[9] http://www.sunstone.com/
[10] http://www.emachineshop.com/