ARCHITECTURAL CONSIDERATIONS IN SDR HARDWARE STRUCTURES

Michael J. Geile, Nova Engineering, Cincinnati, OH, USA; mgeile@nova-eng.com)

ABSTRACT

One primary value of a Software Defined Radio (SDR) stems from its ability to quickly envelop new features and techniques. However, the methods for inducing these features can be severely hampered if the hardware constraints preclude efficient target migration. This paper discloses critical architectural considerations that have been observed in Nova Engineering's experience with targeting waveforms to SDRs. This paper suggests critical improvements in tools and ways in which future SDR suppliers can substantially reduce development and integration costs.

1. INTRODUCTION

Nova's experience on various SDRs has identified a set of design aspects that may benefit future implementers. Nova has observed that approximately 70% of the code/integrate/debug portion of the design cycle is spent in debug. The objective is to decrease this imbalance for future designs and thereby reduce non-recurring costs and enhance maintainability. The issues reported in this paper were recognized empirically and are for informational purposes to enhance future SDR maturity. Some of the approaches are achievable with existing technology while others require breakthroughs in the market.

Figure 1 describes the topics that are briefly addressed in this paper. The circles represent standards or developer actions. A hardware standard is proposed and the discussion shows its importance in containing development costs across the industry as companies adopt the SDR processes. The rectangles itemize the features that bear consideration as the SDR design is formulated.

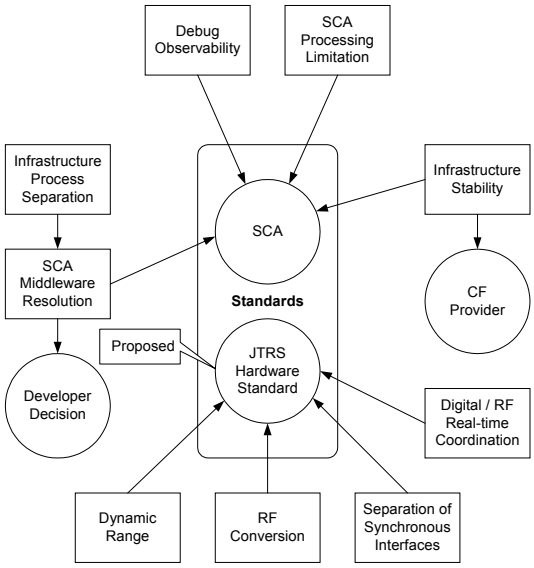

Figure 1: Areas recognized in this paper to reduce SDR implementation risk — circles represent entities, rectangles represent different aspects of the SDR implementation, and arrows show dependencies.

2. JTRS STANDARD IMPROVEMENTS

A primary area for improvement, identified in "Positioning the SDR Business Pendulum – The Case for an "Honest Broker" (to be presented at the November 2002 SDR Forum Conference) is the development of a hardware standard. This standard is proposed to specify the structure of the hardware including real-time interfaces and should be supported by a common set of features required by the waveform applications that are envisioned (present and projected). Various vendors are attempting to independently collect the set of common requirements. A cooperative, unified list would produce a more complete result. The platform standard could overcome many of the limitations that each company individually addresses and would thereby reduce recurring costs. Furthermore, a library of waveforms with design support would reduce development costs with regards to issues such as dynamic range or contributions from various system parasitics (e.g., quantization noise or spurious frequency content). These are costly parameters to derive on a company-by-company basis.

3. TEST OBSERVABILITY

SDR implementations do not naturally support debug observability given that the units require compact mechanical formats and some performance measures can only be completed with the units fully intact and without extraneous connections.

A unit that is not prepared for debug and test can create a significant burden for the system integrators. As stated in the Introduction, about 70% of the code/integration/debug cycle is typically consumed by the debug process. Some of the cost can be attributed to lack

Proceeding of the SDR 02 Technical Conference and Product Exposition. Copyright © 2002 SDR Forum. All Rights Reserved

of rigor in testing key subsystems in a complete manner in an alternate domain that supports efficient debug. Another factor is the limitations imposed due to lack of test planning during the SDR design cycle.

Non real-time debug is an essential step to reduce the cost of this process. For example, a PC emulation of the SDR system can be constructed to execute the final waveform application code with the intent to resolve most of the bugs before accomplishing a port to the final real-time target. The PC environment contains mature tools since it is driven by a large market.

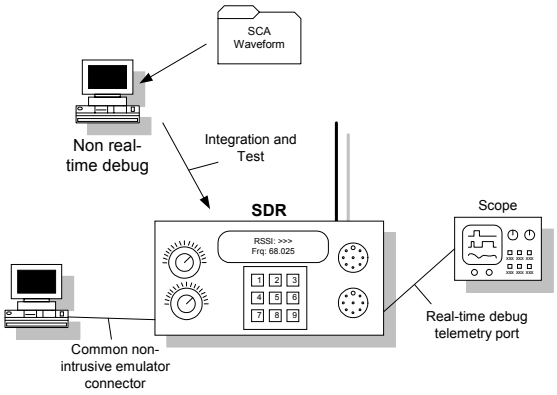

Figure 2: Debug on the SDR is very costly — it is currently feasible to add a non real-time debug step (integrating the final source code) and to add access to debug telemetry. Common emulator support is also a desirable feature.

The ability to execute an emulator on a distributed basis represents a major break-through in such an environment. A common emulator "system" would coordinate all reconfigurable components. Moreover, such a common approach would reduce the learning curve and associated training costs across the entire industry. In current SDRs, the emulator is not a very effective tool since only one processor is controlled while others continue to execute. In many circumstances, the processors that continue to execute either enter an exception condition or lock up due to an unanticipated circumstance. An ideal system would cause all processors to be held in a common state during a breakpoint so that the interactions between processes could be examined. A generalized emulator application that controls all reconfigurable elements including the general-purpose processors, DSPs, and FPGAs is a distant prospect. A common interface that supports just the processors via a port on the SDR may be feasible; several vendors have demonstrated parallel processing tools that might be considered when selecting an SDR approach.

Real-time non-intrusive observability is another important aspect of the design. Casual observation of an indicator of subsystem operations can reduce countless hours of investigation. This is common practice on individual circuit cards and is suggested to carry through to the SDRs in general. One could envision a hardware port on the SDR with pins that are connected to the various processing elements. A test device could be attached to monitor metrics for each reconfigurable element. Even signature metrics could be displayed in real time. For example, Nova has applied PWM techniques via a digital pin to reconstruct an eye pattern for a FSK waveform. As a result, the developer can view the quality of the detected signal using a common scope probe.

4. SCA MIDDLEWARE RESOLUTION

Nova has observed that the CORBA application resolution is inconsistent between vendors. The variations are driven by a tradeoff between reuse and overhead. By restricting the CORBA application to a high level API, visibility into the details of the implementation and the portability of the supporting DSP or FPGA objects is reduced. However, CORBA is not available on many processors and the development cycle cannot sustain the costs to push the middleware down to more primitive elements. Careful attention should be given to the resolution at which the middleware is applied. However, this must be balanced with the ability to deliver portable, open software to the government.

5. SCA MIDDLEWARE PROCESSING LIMITATION

The primary industry targets that support CORBA are Power PCs and Intel processors. Other commercially available implementations of CORBA to alternative processors are rumored but unsubstantiated at the time of this paper. Thus, the SCA inherently constrains the hardware design.

6. INFRASTRUCTURE / PROCESSING SEPARATION

The SCA requires supporting infrastructure to establish common services between the objects. The infrastructure defines a level of complexity while the waveform applications induce additional complexity into the object. Good software practices indicate that two separable yet complex aspects should be tested independently and then integrated after they are tested. Moreover, factoring the processing software from the infrastructure software reveals non-CORBA resources that the waveform "owners" will want to port to future SDRs.

7. INFRASTRUCTURE STABILITY

SCA infrastructure software should be validated as compliant and stable prior to integration activities. A

Proceeding of the SDR 02 Technical Conference and Product Exposition. Copyright © 2002 SDR Forum. All Rights Reserved

validation process currently does not exist. As a result, the infrastructure can contain substantial deficiencies and severely reduce the productivity for a large population of developers since all applications are designed to operate using the infrastructure software.

8. REAL-TIME DIGITAL / RF COORDINATION

Some waveforms require synchronization of digital processing with the RF stream. A support mechanism should be supplied for the transmit and receive signal paths. In some implementations, the reconfigurable processing elements are loosely coupled (at the sample level) to the synchronous RF interface timing. For example, a frequency hopping transmitter may build the waveform in a DSP and requires RF frequency tuning events to be aligned with the waveform as it emerges from the DSP and enters the RF domain. A similar issue exists for the receive path.

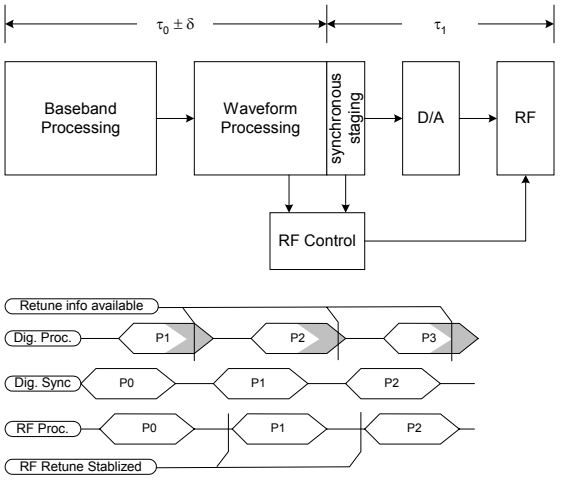

Figure 3: An FH example is shown — the Baseband and Waveform Processing (Dig. Proc.) create the retune information on timing marks with substantial variation. The synchronous digital processing (Dig. Sync) feeds the RF Processing. The retune event must be synchronized with the data stream by a well-define process.

9. SEPARATION OF SYNCHRONOUS INTERFACES

For continuous data operations (in contrast to burst mode), the data and I/Q sampling must operate synchronous with the external data stream. In a DCE configuration, the network clock is passed to the radio by external user equipment. In a DTE configuration, the radio provides the network clock. In either case, the I/Q waveform indirectly represents the data and must be synchronized with the network clock. There are various methods and tradeoffs to be considered. As a general rule, feedforward processes are the easiest to support (but most processing intensive) while analog adjustments are the least reliable.

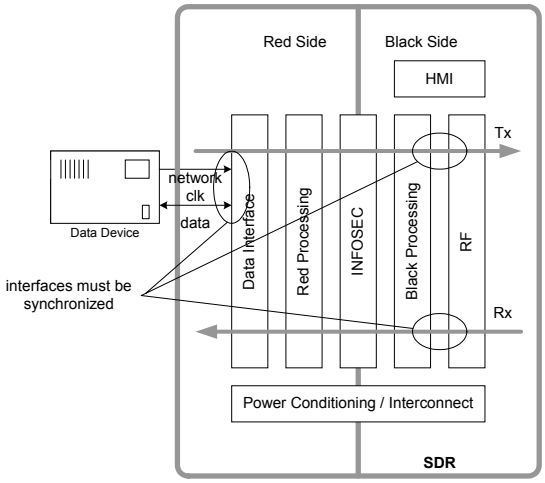

Figure 4: The SDR architecture separates the synchronous interfaces — the selected approach must accommodate a broad range of legacy (e.g., DAMA) and proposed (e.g., WNW) data rates.

10. RF CONVERSION

There are numerous circuit topologies that can be applied to translate information to an RF waveform format suitable for transmission and recovery at the RF band of interest. The methods include analog modulation and frequency translation, direct conversion, digital bandpass conversion, and RF oversampling. Nova has implemented direct conversion and digital bandpass sampling schemes in preceding OFDM designs and in SINCGARS implementations. Our conclusion has been that the parasitics associated with the direct conversion devices (DC offsets, differential gain and phase offsets) available in the market today make it difficult to support the waveform objectives of the SDRs. The parasitics (i.e., single sideband suppression and DC offsets) are a performance limitation for the OFDM implementations. Some frequency hop realizations also encounter significant frequency / temperature dependent affects when the received signal is near sensitivity. Compensation is a reliability burden under these circumstances.

A bandpass approach mitigates the parasitics of a direct conversion system by creating a digital IF. Reconfigurable components can be applied to perform the conversion to/from the bandpass realization. Nova has applied an FPGA implementation to both transmit and receive paths that supports bandwidth from 25 kHz to 1.8

Proceeding of the SDR 02 Technical Conference and Product Exposition. Copyright © 2002 SDR Forum. All Rights Reserved

MHz. A data sheet on this component is available upon request.

11. DYNAMIC RANGE

Emerging SDRs are specified to support a broad set of bandwidths (5 kHz to 10 MHz) and power ranges (greater than 0 dBm to less than –110 dBm) over a diverse operational frequency ranges (2 MHz – 2 GHz). Dynamic range is optimized when an analog filter bandwidth is matched to the received signal bandwidth since out-of-band interference is rejected. Typically, a bank of analog filters are applied that are matched only to a subset of the waveforms. This represents a suboptimum solution over the combination of anticipated set of signal bandwidths. A cost effective reconfigurable analog filter component represents an ideal solution but does not exist in today's market.

12. CONCLUSION

This paper has provided a list of issues that can substantially enhance performance, reduce the design cycle time, and contain costs associated with producing an SCA compliant SDR. The general set of risks should be maintained on a cooperative basis and periodically updated to help future developers avoid the pitfalls experienced in early SDR projects. Some of the suggestions in this paper are currently realizable while others will require the government and industry to invest in tools that will improve future SDR implementations.

Proceeding of the SDR 02 Technical Conference and Product Exposition. Copyright © 2002 SDR Forum. All Rights Reserved