The Killer App – Combining Embedded Processors, FPGAs and Smart Software

Alan Cooke
Curtiss-Wright, Dublin, Ireland, acooke@curtisswright.com,

ABSTRACT

In this paper, the benefits and advantages of combining advanced embedded processing capabilities with an FPGA based approach within a Data Acquisition Unit (DAU) are discussed. The paper begins with a discussion of some of the services and functionality that such a system enables. Basic features such as system discovery, verification, configuration and upgrade are discussed in addition to other value added services such as continuous built in test (CBIT) and embedded real-time parameter quick-look. Finally, the paper discusses some advanced services that could be deployed to these systems such as emerging communication protocols, multimedia connectivity and discovery, and advanced Machine Learning based systems diagnostics.

Key words: FPGA, Embedded Processors, Hybrid Approach, Flexibility

1 INTRODUCTION

Historically, in the Flight Test community CPUs and FPGAs have been the workhorse technologies employed in Data Acquisition Units (DAU). Both technologies have their advantages and disadvantages but their utility have traditionally been constrained by the limited resources available to embedded systems. However, over the last few years the amount of persistent storage, working memory and processing power available to embedded devices has increased exponentially. This has opened up the possibility of deploying ever more complex and intelligent software based services on to the DAUs themselves. This paper describes the benefits of using a mixture of both FPGA and CPU based technologies in conjunction with flexible software in a Flight Test context.

The discussion will begin by describing some of the advantages and disadvantages of both CPU and FPGA based systems. The different types of services that run on a DAU are then outlined in addition to their level of criticality and importance in the context of a flight test. The suitability of FPGAs and CPUs for each of these services is also discussed.

Finally, the possibilities presented by the increase in resources available to embedded systems are then outlined. Specifically, the potential of embedding smart software on Data Acquisition Units is described with particular emphasis on potential services that are not routinely available at present.

2 COMPARISON OF TECHNOLOGIES

Broadly speaking, there are three different approaches taken in production data acquisition systems. Table 1 summarizes the differences between the CPU, FPGA and Hybrid approaches

Technology	Advantages	Disadvantages
CPU	Flexibility Faster Development	Slower boot times Less deterministic More complex
FPGA	Live on Power-up Deterministic	Less flexibility Less performant
Hybrid	Flexibility Performance More deterministic	More complex

Table 1 - Comparison of CPU, FPGA and Hybrid approaches

CPU based systems

The undoubted advantage of CPU based system is the flexibility that they afford. All services are written in software and can be changed and upgraded much easier than FPGA based code. Additionally, the software can take advantage of CPU based functionality like timers, interrupts and specialized mathematical routines. Often there are also off-the-shelf libraries that can be used (for example a UDP/IP stack) which may reduce the overall costs of development.

On the downside, the very flexibility of these systems can make testing and verification a difficult task. Furthermore, it can be much harder to write safe and deterministic code when inherently asynchronous techniques such as device interrupts are used in a design. These systems also have a minimum boot time which may be critical in some applications were brown-outs and intermittent power loss are a concern

FPGA based systems

FPGA based systems have at least two key advantages over the other systems. The first is their determinism and reliability. These systems do not typically respond to asynchronous events such as interrupts and, when written correctly, obey the "works once, works always" rule in that the system will always behave in the exact same predictable way once running. In the unlikely event that FPGA based code gets into an unanticipated state (e.g. as a result of power loss) then it will recover in the next acquisition cycle

The second advantage is their ability to start running (or start running again) immediately after power up or restoration or power. This feature is essential in those applications where brown outs and intermittent power loss is likely.

Hybrid FPGA and CPU Systems

Some vendors also choose to augment FPGA based services with a CPU. Typically the services delegated to the CPU are limited to those where either the gain in implementation flexibility outweighs the benefits gained from the inherent reliability of FPGAs, or to run non-critical services. For those services that do run on the CPU an Operating System is often not used, and instead, the services run on embedded firmware on the CPU itself. The advantages of this approach are that most of the benefits of FPGA systems are retained (such as live on power-up, determinism and so on) along with some of the benefits of CPU based systems (such as flexibility, code re-use).

In general, they still suffer from some of the disadvantages of CPUs such as slower boot times and being potentially non-deterministic. As a result, typically only non-critical services (such as networked based features like SNMP and ARP) are deployed on the CPU.

Constraints

In general, irrespective of the approach taken, the technology has typically been constrained by the resources available in the embedded device. Specifically, the limited amount of working memory (i.e. RAM), persistent storage (i.e. disk space) and processing power (both the number of cores and clock speed) has prevented more complex software in particular from being deployed in DAUs. This was not so much an issue when the volume of data acquired in a flight test is small but modern Ethernet based system on large aircraft can now acquire many gigabytes of data during a test flight.

3 SERVICES HIERARCHY

The services that run on a DAU can be broken into a four layer hierarchy (see Figure 1). At the bottom of the hierarchy are the most important mission critical services, those ones that must exist on a DAU, and at the top are the least important services that are not essential in a data acquisition system.

Mission critical services

Mission critical services can be defined as those that, at a minimum, must be running on a DAU during a test flight. These services include the basic functions of *acquiring* data and either *transmitting* and/or *storing* the acquired data. Another service that is also critical to a large number of applications is *time synchronization*. This is common in large Ethernet based data acquisition networks consisting of many DAUs where the real time correlation of acquired data is a requirement.

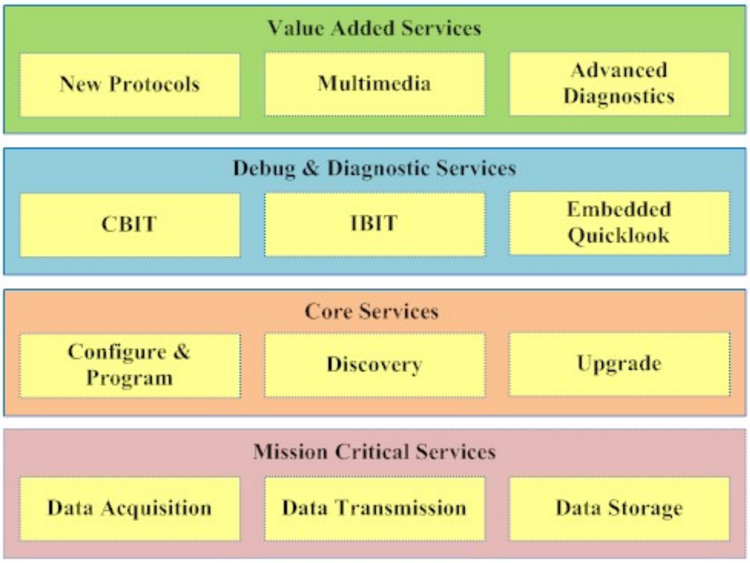

Figure 1 -DAU Service Hierarchy

Core services

These services do not normally run during a test flight but are required in order to create a functioning data acquisition system. These services include those responsible for the setup and configuration of both individual DAUs and the acquisition network as a whole. Arguably they also include services such as firmware and software upgrade, and device discovery.

Debug & Diagnostic services

Whereas both the Critical and Core services are pre-requisites to a fully functioning system, Debug & Diagnostic services can be seen as “nice-to-haves”. They typically are used to trouble-shoot and debug problems that occur in a system. These services include both initiated (IBIT) and continuous (CBIT) built in test and mechanisms for allowing the FTI engineer to view data in real-time.

Value added services

These services are those that are not required for the successful functioning, setup or trouble-shooting of a data acquisition system. Due to the resource constraints of embedded systems these services have traditionally be deployed off the DAU and implemented in software. Examples include complex algorithms for determining the scheduling of data transfers over acquisition networks, advanced system discovery features and pre-flight diagnostics.

4 MAPPING SERVICES

Given the four level service hierarchy presented above it is possible to suggest which technologies are best suited to a given level.

Mission critical services

For the mission critical services such as data acquisition, storage and transmission the key characteristics would seem to be *reliability* and *recoverability* [1]. Whatever else fails, it is vital that data should continue to be acquired and either stored or transmitted for later analysis. Ideally, data acquisition should be continuous, but in the event of intermittent but unavoidable power loss, the downtime of the system should be minimized.

Both of these requirements imply that FPGA technology is the clear winner. It has the inherent advantage of being both more reliable and deterministic compared to a CPU based system. Additionally, in the event of a power disruption, it can recover immediately.

It is of course possible to implement mission critical services using pure CPU based technology but this task is intrinsically more difficult. The effort in verifying and validating the software that is running on the CPU is complex, time consuming and error prone. It often requires the rigorous and continuous application of strict design rules (such as a ban on using interrupts for certain tasks). This problem is multiplied many times more if the software runs on top of an operating system where task scheduling and resource allocation are usually either out of the hands of the programmer or difficult to influence. Use of a specialized Real-time Operating System (RTOS) can be used to ameliorate these problems but they are generally more expensive than, for example, a customized embedded Linux system and thus can add to the overall development cost and the cost to the end user. Furthermore, boot time is generally many times longer than for FPGAs.

Core services

As discussed, while these services are not a requirement during a test flight they are needed in order to configure a system. While in theory it may be possible to configure a system using FPGA based designs, in practice this requires the use of software.

At first glance, these requirements appear to favour the CPU based approach. Software can easily be written to configure a DAU and then deployed directly onto the DAU and perhaps exposed to the FTI engineer via an HTML based interface in a browser. Discovery can also be deployed directly on the DAU, and again, presented to the engineer through a HTML based interface in a browser.

However, this functionality can also be implemented in software running on separate devices such as a laptop or desktop computer. Typically, configuration involves two steps. First the system is defined in a vendor provided application and then the system is configured by connecting to the DAU and uploading the configuration to it. In order to do discovery, the laptop or desktop computer needs to be connected to the DAU and a vendor provided "Discovery" tool is used to determine the configuration and make-up of the DAU.

Where the pure CPU based approach faces challenges is in large data acquisition networks consisting of perhaps forty or fifty time-synchronized DAUs. Depending on the requirements, this may necessitate the scheduling and delivery or large volumes of data in specific time windows to multiple destinations. The complexity and resource requirements needed to perform this task can typically be more than is traditionally available in a single DAU. Furthermore, such algorithms usually require a high-level overview of how the system is connected together, the sample rates for each DAU and the delays through each node and switch in the acquisition network. In practice therefore, these types of services need to be run off the DAU.

Debug & Diagnostic services

Once a data acquisition system has been configured it is usually necessary to verify that the system is behaving as expected by running a suite of tools designed to be run either in the lab or on the aircraft itself. These tools can include CBIT and IBIT, Auto-shunt and Auto Balance and the inspection of real-time data coming from DAUs.

Both CBIT and IBIT can range from simple tests such as parameter threshold checks to arbitrarily complex ones. Depending on the level of complexity, this functionality can be implemented in either FPGA or CPU. For complex testing beyond standard BIT, CPUs have the advantages of flexibility and so on described earlier. While for simpler implementations of for example, CBIT, the FPGA may be more advantageous if the data is required to be transmitted with the rest of the test flight data.

Value Added services

These types of services tend to be more complex than services in the other layers, and because of this, typically must be run away from the DAU. These services include advanced diagnostics involving complex algorithms and software for downloading and viewing recorded data (for example video data).

5 ADVANCED SERVICES

As discussed, the resource constraints traditionally imposed on embedded systems have meant that it is usually only the mission critical and core services that run on a DAU. Some vendors have also deployed basic debug and diagnostics services on the DAU but these are generally limited in complexity.

This section discusses some of the potential services that could become available on the latest generation of DAUs[1] that take full advantage of the possibilities made possible by Moore's Law[2]

Discovery

In today's FTI environment discovery is typically confined to describing the very basic characteristics of the DAU and also perhaps to describe to what other DAU's or network nodes it is connected. This information would commonly include serial numbers, firmware and other version numbers and perhaps some information on how the DAU is configured. With the deployment of smart software it is now possible to move beyond these basic features.

The first obvious enhancement is to make each DAU and data acquisition network fully self-describing. This essentially involves asking each DAU four questions

- What are you?
- How are you configured?
- How can I configure you?
- What does the network that you are part of look like?

[1] Such as the Axon from Curtiss-Wright

[2] https://en.wikipedia.org/wiki/Moore%27s_law

This can be achieved in two ways. The first is to provide an embedded user interface on the DAU and present this information in a user friendly and interactive way. The second is to provide a publicly accessible API[3] that would allow either vendor provided or user developed software off the DAU to perform the same function as the embedded UI[4].

In addition to the information traditionally provided by vendors, it would include the full meta-data used to configure the *entire system*, when it was configured, who configured the system and any other information deemed pertinent by the user or vendor. Furthermore, the APIs would also be fully self-describing, allowing users of the equipment to quickly understand how to integrate with and interrogate the DAU and the acquisition network.

Setup & Configuration

It is now possible to move beyond the static and simplistic browser based user interfaces that have existed heretofore and to embed a browser-based GUI in a DAU as sophisticated as anything found on a desktop or laptop computer, including interactive wizards and intelligent agents.

Using the latest responsive web design techniques[5], frameworks and tools it is easy to write browser based user interfaces that adjust what is displayed to the user based on the size and type of the display being used. The content presented to the user can be adjusted so that it fits comfortably in smaller screens such as mobile phones and tablets by selectively displaying and removing different GUI elements and by resizing others.

In addition to this, it is also possible to use the same code base used in the browser based GUI to create cross platform Apps that can run on iOS, Android and other mobile platforms(See Figure 2).

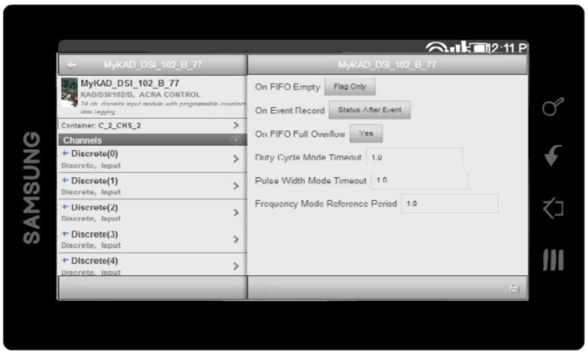

Figure 2 - Tablet based configuration

Petabyte test articles

Most contemporary DAUs are limited to acquiring only several megabits of data. With the advent of the latest generation of DAUs this ceiling has been removed and it is now possible to acquire and process many gigabits of data in a *single* DAU. These Gigabit class DAUs open the door to acquiring data from multiple high-speed and high bandwidth data sources *concurrently*. For example, they permit multiple high-speed and high definition video sources to be fed into the *same* DAU. These Video sources could also sit alongside other high volume data sources such as a High Speed Avionics Bus, multiple high bandwidth analog signals, and 1 and 10 Gigabit Ethernet data sources.

As these next generation DAUs become more widely deployed it will eventually result in many hundreds of Gigabits (or Petabytes), of data being acquired continuously on an aircraft.

---

[3] Most likely a RESTful API (see https://en.wikipedia.org/wiki/Representational_state_transfer)
[4] The Axon from Curtiss-Wright provides both of these methods
[5] See also https://en.wikipedia.org/wiki/Responsive_web_design and http://www.w3schools.com/html/html_responsive.asp

This will be especially so for large aircraft where there can be easily fifty or more DAUs on an aircraft.

This volume of data, and the number of DAUs and other nodes on the data acquisition network, pose significant challenges for FTI vendors. Specifically, determining how data is synchronized and how data transfer is scheduled on an acquisition network becomes much more complicated. It requires specialized configuration software that is intelligent enough to model delays at each node in the network, can take into account the time constraints of the ultimate destination for the data (e.g. a recorder versus a IRIG-106-Ch4 transmitter), identify potential data bottlenecks in the system and is flexible enough to work with any size or type of data acquisition topology the FTI end user can throw at it.

Data extraction and Inter-connectivity

Given the large volumes of data generated as part of a test flight it is now almost unheard of for most, if not all, of the data not to be stored on board and then extracted post flight for analysis on the ground. Several standards (most notably the Chapter 10 standard) have been developed in an attempt to unify the data extraction process, which has promoted varying degrees of compliance. However, the sheer volume of data lends itself to analysis using modern “Big Data” techniques.

The latest generation of DAU facilitates the possibility of leveraging modern consumer and industrial standards and protocols for downloading and distributing data reliably for post flight analysis. These protocols include the DDS[REF XXX], MQTT[REF XXX] and AMQP [REF XXX]. These are widely used and robust standards with large associated toolsets and vendor implementations.

Any one of these protocols could be supported natively on a next generation DAU and used to download and distribute the (potentially huge volume of) acquired data into the end users data analysis systems and data distribution backbone.

Debug & Diagnosis

Most DAUs provide a very limited set of debug and diagnostic tools (if they provide them at all). As discussed, these typically include CBIT, IBIT and shunting tools. Again, the latest generation of DAUs can offer much more.

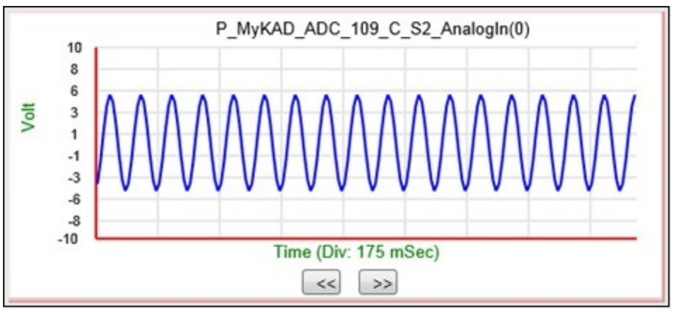

Figure 3 - Embedded real-time display

The basic debug & diagnostic tools can be extended to include sophisticated, system-wide tools. For example, software on the DAU can examine a configuration and warn of potential data bottlenecks in the acquisition network. The DAU can also come with an embedded real-time data viewer (see Figure 3)[6] the can be used to view key parameters/signals of interest on the aircraft.

[6] For example, the Axon from Curtiss-Wright features an embedded real-time data viewer

Other more sophisticated diagnostics are also beginning to be deployed on DAUs. For example, Curtiss-Wright as part of their *Machine Learning for Advanced System Diagnostics* project, are experimenting with deploying Machine Learning algorithms on their DAUs to detect and classify problems with sensors and wiring in the data acquisition network and also to detect and flag anomalies as they occur in the system. These are especially useful in pre-flight but also for flagging problems as they occur during a test flight

Recent trends in Artificial intelligence and Machine Learning also open up the possibility of deploying services that actively aid the FTI engineer to construct, verify and troubleshoot their configuration. For example, the software may warn an engineer that the throughput of a particular node in the configuration may exceed the bandwidth available and suggest alternative strategies.

THE KILLER APP

Given the above discussion, it is suggested that the winning combination of FPGA, CPU and Smart Software have the following attributes

1. Reliability and Recoverability. Data acquisition is the core function of an FTI system and must meet the highest standards of reliability and resilience.
2. Provide an embedded Setup and Configuration user interface comparable to that experienced on desktop and modern mobile platforms. It is important to be able to define and make changes to a configuration simply and quickly, either on or off the aircraft. A "Whole Picture" view of the acquisition network is also essential for the latest networked systems
3. Advanced debug and diagnostics capabilities. This functionality can potentially save large sums of money over a flight test campaign. Intelligent diagnostics can potentially find issues before an aircraft takes to the sky
4. The ability to ingest multiple gigabits of data on one DAU and to fit seamlessly into "petabyte" test systems. The amount of data acquired as part of a flight test is only going in one direction.
5. The ability to integrate with modern big data and legacy infrastructure. Large volumes of data necessitate the ability to download data from test articles and to distribute it easily for post processing and analysis. However, small systems and legacy infrastructure will also need to be supported.

SUMMARY

This paper started by discussing the technologies historically used by Data Acquisition Units in flight test and outlined and categorized some of the services employed in these systems. The benefits of FPGA versus CPU based services were also discussed. The paper then focused on the limitations imposed by the traditionally resource constrained DAU. Typically, it is only the data acquisition and transmission data services, in addition to some limited setup and diagnostic services that run directly on the DAU while other more advanced functionality is usually implemented in software that runs off the DAU.

The paper then discussed some of the possibilities facilitated by the latest next generation DAUs that are largely free of the constraints that exist on earlier DAUs. These possibilities include embedded desktop-like browser based configuration and setup, Advanced Diagnostics, the ability to ingest Gigabits of data and seamless integration with data processing infrastructure.

REFERENCES

[1] D. Buckley, "The Challenges of Data Acquisition in Harsh Remote Places," in *Proceedings ITC 2015*, Las Vegas, 2015.