WHITE PAPER

How to select Space processing devices for in-orbit edge computing applications?

December 2022

TELEDYNE e2v | Semiconductors
Everywhereyoulook™

ABSTRACT

This white paper discusses the **challenges and demands** of **in orbit edge computing** and **how they can be handled with processing devices.** The paper outlines the **different processing approaches** which can be used to address the challenges in orbit before **presenting several case studies** which demonstrate the use case for **the different processing methods and the selection for each one.**

DEMANDS AND CHALLENGES OF SATELLITE PROCESSING

Satellite applications have a significant impact on modern life. From the more obvious satellite navigation to precise timing used for global banking systems and telecommunications, in addition to earth observation systems which monitor the health, weather of our planet, providing in some cases early warnings, satellites are used every day around the globe. The satellites that support these applications operate in a variety of orbits from LEO to GEO. They also have widely varying sizes and lifetimes, from the smallest CubeSats in LEO expected to operate for one year with a weight of 1Kg, to large telecommunications satellites operating in GEO weighing 6500 Kg with an expected operating lifetime of 15 years.

What makes all these satellites possible is processing capability. Processing capability also forms a key element of getting the satellites to the correct orbit, as processing capability is used by the launcher to ensure correct and safe operation along with payload deployment.
Regardless of how this processing capability is implemented, whether in a microcontroller, microprocessor, FPGA, or ASIC, it makes the satellite to be able to control attitude, communicate with ground stations, manage the satellite payloads, and most crucially, implement the payload functionality.

This processing capability within a satellite is split between two distinct elements: the platform and the payload. Equipment within the platform is responsible for the management and operation of the satellite itself. This includes communications, attitude control, thermal management, and power management etc. Conversely, payload equipment implements the mission of the satellite. This may be telecommunications, earth observation, or even interplanetary missions. Processing within the platform is implemented to achieve the desired functionality of the payload like image processing, camera interfacing, or telecommunications beamforming for example.

Of course, when processing applications are deployed within space applications, they face a much more challenging environment than that which is found in many terrestrial applications. These processors must be able to withstand the effects of temperature, shock, vibration, and radiation while still achieving the desired reliability.

Each of these challenges affect the processing element in different ways. A wide operating temperature can affect the performance of the processor because devices typically operate faster at low temperature and slower at high temperatures. Temperature can also affect the start-up conditions at low power where more current may be required. At high operating temperatures, the higher power dissipation can impact the derating requirements.

Shock and vibration are mainly thought of as physical failures. However, vibration and shock could have short term impacts on system operation if connections or connectors are impacted.

The main consideration for the in-orbit processing capability is the radiation performance. Radiation in orbit takes two main kinds: Total Ionising Dose (TID) and Single Event Effects (SEE). Total Ionising Dose is a long-term effect in the degradation of the component electrical parameters over time. Typically, TID can impact the timing and power dissipation of the device and sometimes by the end of life, the device performance is degraded and power dissipation increased. There is one unique and very interesting element of TID which is called Enhanced Low Dose Rate Sensitivity (ELDRS) where the device exhibits more parametric shift when the TI dose rate is low. ELDRS affects mainly bipolar and BiCMOS technologies.

The second type of radiation effect, known as Single Event Effect (SEE), is perhaps the more commonly thought about. SEE results in bits or multiple bits flipping in the processing memories. However, SEE can also be destructive in some technologies, it is critical for space applications to use components which do not exhibit destructive latch-up (SEL).

There are steps which are taken by the developer to mitigate these effects in devices used for space processing. The most common approach is to select a device qualified for use in space applications for example QML V or QML Y for space for integrated circuits. Traditional space applications often have requirements on component quality which can lead to considerable costs and long leadtimes. New space applications often deploy commercial components which are not necessarily qualified but have some heritage from either testing or being used on similar mission profiles. In some cases, new components with no heritage are used however, this of course introduces risk to the program.

Processing solutions for space range widely and include the following:

- **Microcontrollers (MCU) – Tightly integrated processing solutions which contain the necessary memory and peripheral interfaces. Typically, microcontrollers offer a lower processing capability.**
- **Microprocessor (MPU) – High-performance processing systems which require supporting devices such as NV RAM, DDR RAM etc. Peripheral interfaces may be provided or may be implemented by an external bridge. Typically, Microprocessors are used where high-performance processing is required.**
- **FPGA – Programmable logic which can be configured to implement processing solutions by leveraging the parallel nature of logic. Examples include implementing an image or signal processing pipeline. Sequential structures can be implemented by using Finite State Machines or by a softcore microcontroller such as the LEON 3FT or NOEL Risc V for complex control structures. Typically, FPGA are used where high-performance processing is required along with a hard real time response.**
- **Devices featuring a mix of the above,**

Of course, each one of these solutions has a different use case and the selection of the most suitable will depend on the application and mission requirements and analysis.

Microcontrollers are used to integrate low-level control and commanding like receiving data over a communication bus and configuring the system in line with that command. Examples include thermal management of an imaging camera where temperature set points are required and PID loop is executed on the microcontroller. With a microcontroller being a tightly integrated solution, the implementation architecture is much simpler and usually requires only power architecture and clocking.

For higher performance applications, both microprocessors and FPGAs can be used to implement the application decision. Microprocessors are an ideal selection when sequential high-performance processing is required in addition to working with high-level frameworks (e.g., AI / ML etc.). FPGAs are ideal for applications requiring high parallel solutions with flexible interfacing or where a highly deterministic and responsive solution is required for safety.

There may be situations where several solutions can fit the requirements of the application for example, a microcontroller or FPGA, or microprocessor or FPGA. In these cases, the selection criteria come down to the institutional knowledge and experience regarding which solution presents the lowest technical and programmatic risk.

One key element that decides the election of MCU, CPU or FPGA is the performance required by the application. To be able to compare processor performance, we can use metrics such as Millions of Instructions Per Second (MIPS) or Floating-Point Instructions Per Second (FLOPS). Typically, these are defined as a specific number of MIPS/FLOPS per MHz enabling the developer to scale and understand the performance for the processor's current configuration. This also enables the performance of multiple cores to be addressed. The same approach can also be used to determine FPGA performance based off the DSP resources within the device. These can provide an indication of the Multiply Accumulate Cycles which can be achieved often and given as GMAC or TMAC along with the indication of the related FLOPS. However, this kind of metrics doesn't translate the capability of one architecture to be more fitted to do a certain type of operations or calculations.

Hence, in addition to the pure processing capabilities, there are several different questions and decision points which can be used to select the most appropriate solution.

1. **Application** – Is the application largely serial or parallel in nature? An example of serial application would be networking / communications and processing of packets or the controlling and configuration of a satellite payload. A parallel application would be telecommunications processing or image processing.
2. **Qualified Environment** – Does the selected technology exhibit the required qualification for the environment for example QML V?
3. **Heritage and Technology Readiness Level** – Does the selected approach have any flight heritage and has the TRL been demonstrated to the necessary level to reduce the technical and programmatic risk? No heritage or low TRL does not prohibit use of the application but it will indicate there is a need for an additional testing to increase TRL.
4. **Design Reuse** – Can modules / IP blocks / algorithms from previous projects be reused, reducing non-recurring development costs?
5. **Tools Chains and Supporting Infrastructure** – Does the institution have the necessary tools and licenses to develop for the potential target device? This would include debuggers, simulation environments, and bitstream / binary licenses.
6. **Power Management and consumption** – Many space applications are power constrained. This can constrain the selection of the processing element depending upon the maximum power required and power management options available (e.g., the ability to shut down the unused peripherals and reduce clocking to be more power efficient).
7. **Interfacing** – Can the data be brought on and off chip?

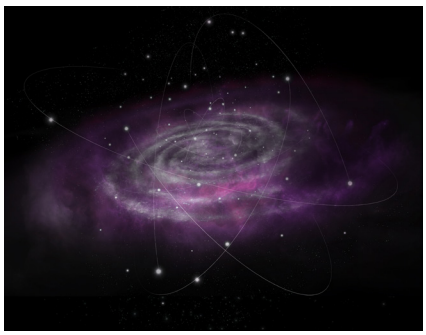
One of the best ways to help make the selection between FPGAs, MCUs or CPUs is to use a weighted **Pugh Matrix**. A Pugh Matrix lists the important parameters and assigns a score to each parameter to determine the most appropriate solution when several potential solutions exist. As we examine the different case studies, each will be presented with its own Pugh Matrix to demonstrate the most critical parameters and selection for that application.

COMPONENT RANGE

There is a diverse range of microcontrollers, microprocessors, and FPGAs available for use in space applications. These range from devices which are fully compliant and qualified for space applications to commercial components up screened for new space applications. Typical suppliers of processors and FPGAs include AMD Xilinx, Teledyne e2v, Microchip, Lattice, or NanoXplore.

CASE STUDIES

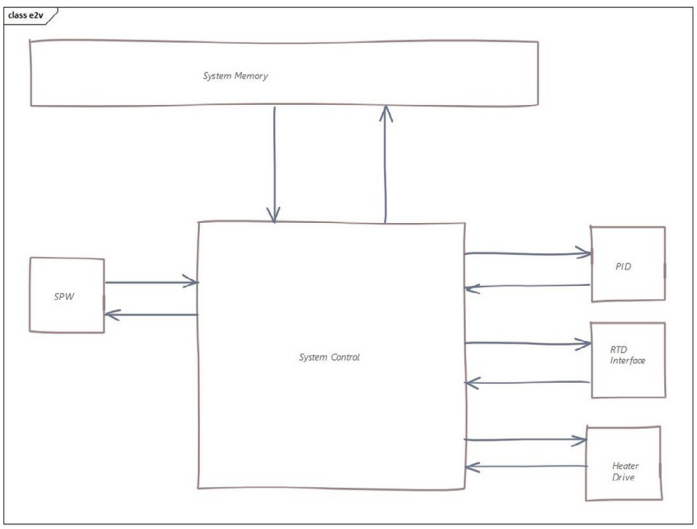
A. THERMAL MANAGEMENT



The first case study is a platform temperature control subsystem. These are typically used on imaging satellites for either earth observation or more scientific universe exploration missions. These payloads are tasked with controlling the temperature of an imaging satellite's optical bed. The system is required to received setpoints and configuration data over a SpaceWire interface. These setpoints and configuration data are then used to sample temperature sensors and calculate the drive power needed to be output by heater elements to keep the optical bed at a stable temperature. The optical bed will contain several Platinum RTD sensors and heater elements. For each heater drive, a PID loop is run which takes the temperature input from one or more temperature sensor. The desired set point temperature and PID constraints will be provided over the SpaceWire interface. The association of heater drives to temperature sensor is also defined over the SpaceWire interface. This provides a very flexible solution. It does, however, mean there are several configuration registers -potentially several hundred. In operation, each of the PID loops run and update the heater drives. Once all the PID loops have been run, the system halts until the next scheduled up time. PID loop update times are typically configurable between five and twenty seconds over the SpaceWire link.

This application is highly serial in that the configuration and control information is defined over a communications interface and then each PID loop is run in sequence. The operations involved in each PID loop are also sequential. The desired number of temperature sensors are read, the PID loop is run, and the updated heater drive is output. Key requirements in this solution include:

1. Implement a SpaceWire interface
2. Ability to store configuration data
3. Implement PID, not time critical and accuracy is more important. High performance is also not critical
4. Interfacing with analog front-end Platinum RTDs – Typically low-speed ADCs are used
5. Heater drives PWM based with appropriate analog scaling – GPIO and PWM elements are critical.



	FPGA	Microprocessor	Microcontroller	Comment
Application	1	3	5	PID Algorithm, large configuration, system management
Throughput	5	5	5	Soft Real Time system
Interface	5	3	5	PWM, GPIO, I2C and SPI Required
Power	3	1	5	Low Power

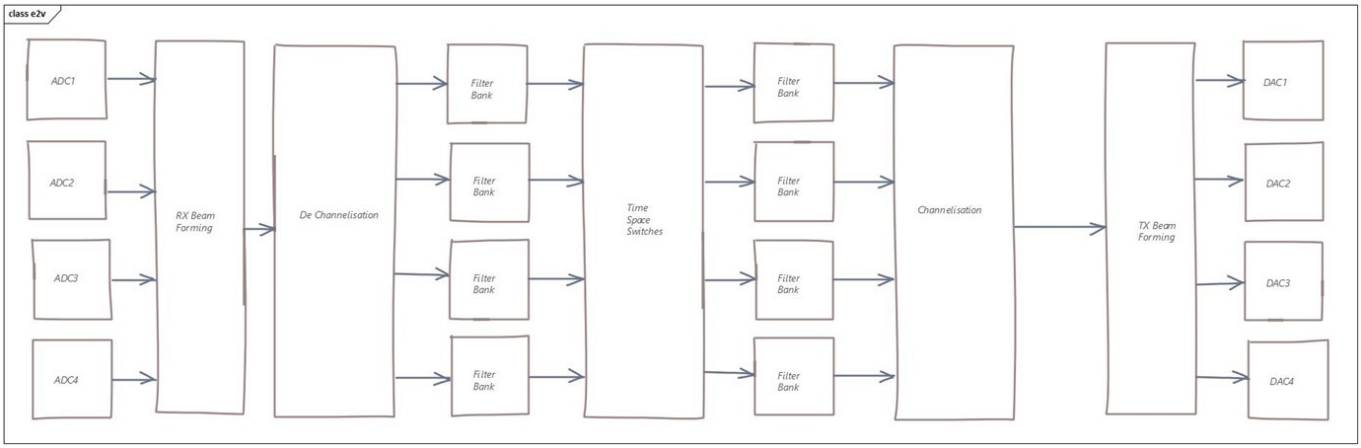
Applying a Pugh Matrix to this solution shows that the correct selection for a mission such as this would be the MCU. The matrix shows, when considering application, throughput, interfacing, and power, the Microcontroller is the best option

The unique interfacing requirement of space wire support means either a specialised MCU or FPGA is used. An FPGA solution would be significantly more complicated to implement a memory mapped, sequential processing. A MCU is ideal for this application, in such a simple application the microcontroller could run using either a bare metal hardware abstraction level framework, alternatively a simple real time operating system such as be deployed to aid with scheduling and sequencing.

B. TELECOMMUNICATIONS

The second case study is a telecommunications processor typically deployed in GEO synchronous orbit and used to provide global civil and military communications. Typically, these telecommunications processors function on the received RF waveforms and no decoding / re encoding is performed. Therefore, these telecommunication processors receive the RF waveform de channelize the waveform, and route the signal as required before channelization and beamforming. This requires the use of high speed, high-bandwidth ADCs and DACs which can receive and generate the signals. Interfacing with these ADCs and DACs will use LVDS or CML technologies to support the bandwidth requirements.
For processing, this requires high performance capable of implementing multiple ADC and DAC interfaces, in addition to performing multiple DE channelization and channelization functions (IFFT and FFT operations).

Telecommunications processors provide the ability to transmit data in both direction to and from spot beams covering areas of globe to a downlink. To achieve this the following architectures are often used.



The Digital beam forming on the receive and transmit channels are required to necessary spot beams. Digital beamforming can be implemented using a N point FFT, the size of the FFT depends upon the number of beam forming elements. While the frequency of operation is depending on the sampling rate, this sampling rate will be between 500 MHz and 1GHz. Post beam forming is required to perform de channelisation to separate each of the individual channels. This is achieved using a Weighted Over Lap Add FFT (WOLA FFT), WOLA FFT can be used to perform both channelisation and de-channelisation. The multiplication rate of the WOLA FFT is ½ BK log2 K where B is the sample rate and K is the number of channels. To ensure there is no inter channel interference filter banks are used on the output of the channelisation process, the multiplication rate of the Filter is 3K2B.

Taking a simple example for sampling rate of 400 MHz and 100 channels, this equates to a WOLA FFT rate of 132 Giga multiplications per second, while the filter banks require significant number of multiplications to perform the channelisation filtering this requires 6 Tera Multiplications per second. There are two WOLA FFTs required (channelisation and de-channelisation), throughout is also highly deterministic requiring all stages to be performed in parallel.

Key requirements in this solution include the following:

1. Support for multiple high-speed ADC and DAC interfaces
2. Highly parallel processing – Multiple channels must be processed at once
3. Possible extension for decoding and regeneration of waveforms to increase signal-to-noise performance
4. Ability to implement beamforming calculations for return path
5. Power efficient processing solution

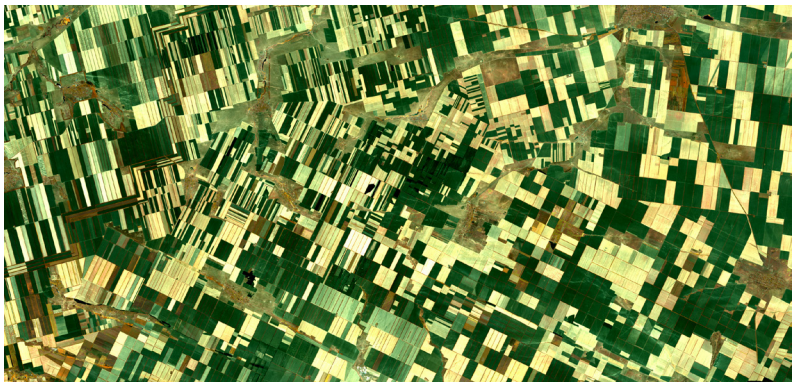
	FPGA	*Microprocessor*	*Microcontroller*	*Comment*
Application	5	3	1	FFT, IFFT, Space Time Switching, WOLA FFT etc
Throughput	5	1	1	Very High Through put, hard real time demands
Interface	5	1	1	Bespoke High-Performance ADC/ DAC Interfaces
Power	3	3	1	Power dissipation will be high, offer good W/MHz processed

For this application, a FPGA would be the correct processing solution, due to the high performance and throughput requirements, modern FPGA can perform up to 22 TeraMACs. While the FPGA can achieve the 6.3 Tera Multiplications required, an Arm® Cortex®-A72 processor will achieve circa 14Giga Multiplications per second, per core at 1.8 GHz. Another reason for the selection is that the I/Os of the FPGA are flexible and capable of high performance, enabling the support of high-speed ADC and DACS. The parallel nature of FPGAs enables the implementation of multiple FFT and IFFT structures and allow multiple channels to be processed at the same time. If one FPGA is not sufficient to implement the solution, multiple interconnected FPGAs can be used to implement the solution using high speed serial links. Although FPGAs can be power hungry, in this application, the FPGA will offer the most efficient FLOPS/Watt for implementation and elements of the design. For example, unused channels in a configuration can be clock gated further reducing the power required.

C. EARTH OBSERVATION SATELLITE

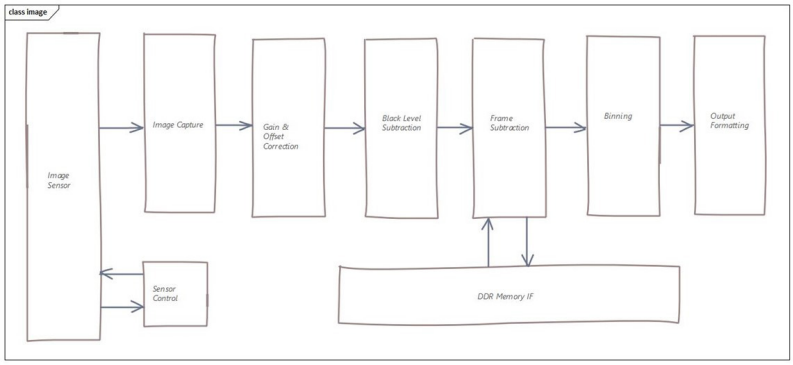
The penultimate case study looks at an earth observation imaging system. This system images the earth as it flies over head. Data produced is considerable, a typical image sensor in an EOS observation scheme will generate 3Gbps. 1000km swath takes approximately 143s producing approximately 428.6Gbit (53.58GB) of data is generated. This data is normally formatted using CML this maybe bespoke or an industry standard such as SMPTE.

Once the images have been captured, they are post processed to assign a coverage rating to the image (e.g., is the image obscured by cloud or weather cover). When imaging, the image sensor will generate many GBytes per second of image data which needs to be captured by the EOS imaging system. Once the image data has been captured and stored in the front-end electronics, these images must be classified for coverage.



Once classified, the image and ranking are passed to a data handling unit. Key parameters for this application include:

1. High performance interfacing capable of receiving high-bandwidth data (e.g., camera data)
2. High performance parallel processing – Reception and formatting of images
3. Ability to implement high-throughput ML inference for cloud coverage ranking
4. Support for high-bandwidth memories for buffering image tiles
5. Parallel processing



	FPGA	Microprocessor	Microcontroller	Comment
Application	5	5	1	Image Capture and Image Processing
Throughput	3	3	1	Hard Real time processing requirement on Image capture
Interface	5	5	1	Bespoke or Standard interface high performance
Power	3	3	1	Power dissipation will be high, offering good W/MHz processed

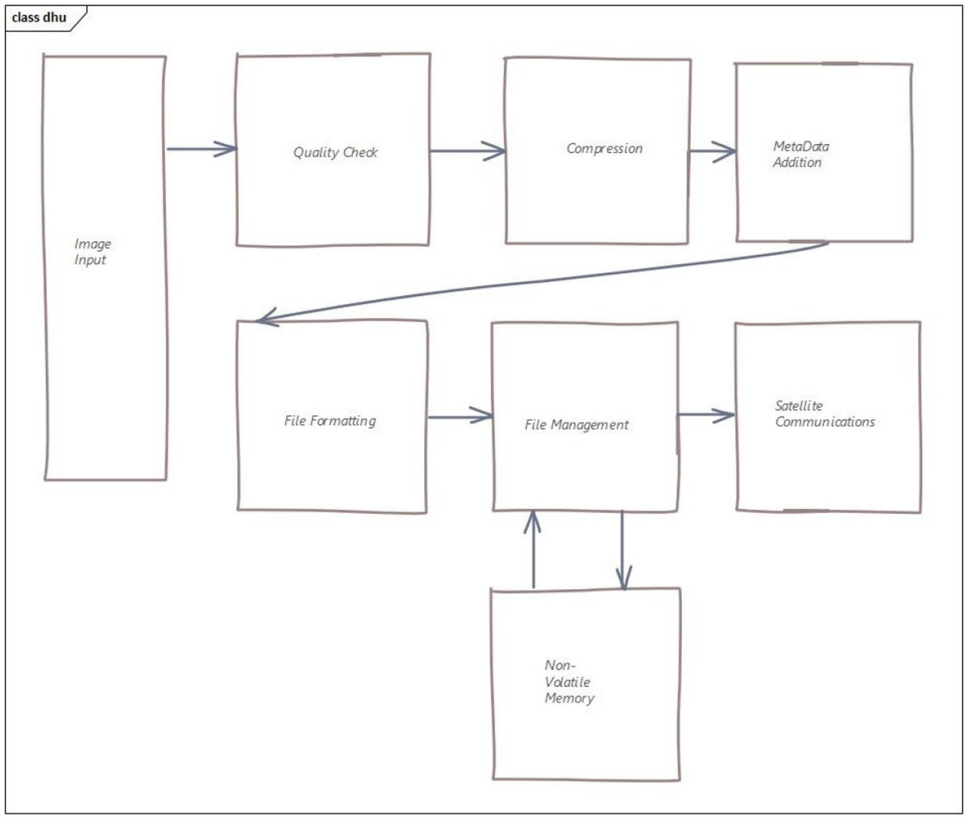
In an application such as this, either a FPGA or a high-performance multi-core processing solution can be utilized to receive the image and run the machine learning inference. Architecting a high-performance multicore processor solution would require the use of cores dedicated to specific tasks, e.g., image reception, image processing and formatting for onwards transmission. Architectural features which support Vector Processing Extensions and Single Instruction Multiple Data (SIMD) can provide significant performance enhancements to help achieve the desired performance. Implementation in a high-performance multi-core solution will also require the deployment of an operating system, one which is capable of abstracting away system level functions. Operating systems such an Embedded Linux or RTEMS are often deployed, but Real-Time Operating Systems can also be used to support these applications if a higher degree of safety is required. Libraries such as OpenCV, can also be leveraged for development of the image processing application.

One of the main advantages of a software-based approach is the development time for the project, which can be significantly reduced compared to what might be the case with an FPGA development. For instance, FPGA development times can be considerable depending upon the regulatory environment e.g., ESA compliance. The development of software applications can leverage functionality provided by operating systems, commonly used libraries etc.

SW development may also be easier to commission and debug on the system as there is greater visibility of the registers, memory, and variables within the scope of software debugging. The iteration time on SW development is also significantly reduced as build and verification time is significantly faster.

Selection of either a FPGA or processor-based solution would depend upon institutional heritage, skill base, and preferences.

D. DATA HANDLING UNIT



	FPGA	Microprocessor	Microcontroller	Comment
Application	3	5	1	Image compression & file management, communication with satellite systems
Throughput	3	5	1	Compression needs high performance,
Interface	5	5	1	Standard High-Performance interfacing used e.g. 10G, PCIe
Power	3	5	1	Power dissipation will be high, offer good W/MHz processed

The final case study is a data handling and processing unit (DHU). In an image processing satellite, the DHU is responsible for receiving images from the front-end electronics (i.e., the camera interface) and performing compression, storing in file system, and recall for downlink upon request. This means the DHU must be able to keep up with the data rate of images coming in during the imaging phase of operation. During the imaging phase, incoming images must be compressed and stored in the file system for later downlinking. Curing the downlink operation, the DHU simply receives the requested image from memory and forwards it on to the RF link.

Key parameters for this application include:
1. High-performance processing to support imaging phase
2. Ability to manage file formatting
3. Ability to perform image recovery from NV RAM system
4. Ability to perform loss less compression with high throughput

Data reception can be achieved from the front-end electronics using standard high-speed interfaces such as PCIe or 10G Ethernet. This assuming a data rate of 3 Gpbs as connected to the front-end electronics, both PCIe and 10G Ethernet provide sufficient bandwidth.

Compression algorithms such as JPEG-LS, JPEG-2000 and CCSDS compliant schemes are used to implement the lossless compression. Assuming a loss less compression ratio of 50% the 1.5 Gpbs of data needs to be compressed using one of these algorithms. A multi core processor such as the Arm A72 can perform up to 14 Giga Flops per core or 56 GFlops for a quad core solution. This level of performance is easily capable of implementing the desired functionality. As the compressed images need to be stored in a file system e.g SAFE file format commonly used in EOS missions an operation system is often required to manage the storage and recovery of image tiles.

This system does require high-performance processing providing a mix of parallel and sequential processing. As a result, the best implementation for this is a high-performance processor which can receive the image tiles and perform the compression. Sequential processing is then needed to communicate with the satellite, manage the file management system, and retrieve and manage files within the non-volatile memory.

CONCLUSION

Satellites require a significant amount of processing and functionality to be able to achieve their missions. When it comes to selecting the most appropriate features, there are several different selection criteria which can be used in conjunction with the end application to determine the most appropriate solution including elements directly related to the implementation including application, throughput, interfacing, and power. Along with the implementation element the selection criteria also rely upon the technology readiness level, company heritage and skill base.
While each application will have different selection criteria one, there can be established an initial decision matrix to provide a granular indication of appropriate technology as a starting point.

	Most Appropriate	*Consider*	*Do not consider*
High performance & Hard Real Time Response	FPGA	Microprocessor	Microcontroller
High Performance Image or AI/ ML	Microprocessor / FPGA		Microcontroller
High Performance Data Handling	Microprocessor	FPGA	Microcontroller
System Configuration	Microcontroller	FPGA	Microprocessor

The key thing for engineers is to have a range of tools in their toolbox to select the most appropriate for the task in hand.

TELEDYNE e2v Semiconductors

For further information, please contact:

Thomas Guillemain,
Marketing & Business Development,
Data Processing Solutions
thomas.guillemain@teledyne.com

TELEDYNE e2v Semiconductors

For further information, please contact:

Thomas Porchez,
Application Support,
Data Processing Solutions
thomas.porchez@teledyne.com