EVS30 Symposium

Stuttgart, Germany, October 9 - 11, 2017

Software bottleneck removal and design issues solving for high frequency control of hybrid and electric powertrain.

Anselme Lebrun[1], Khaled Douzane[2]

[1]*Anselme Lebrun, 535 route des Lucioles 06560 Sophia-Antipolis, France, anselme.lebrun@silicon-mobility.coml*

[2]*Khaled Douzane, 535 route des Lucioles 06560 Sophia-Antipolis, France, khaled.douzane@silicon-mobility.com*

Short Abstract

Performance improvement of hybrid and electric powertrain is directly related to energy efficiency increase, battery range extension and battery charging time reduction. Improving this performance requires more sophisticated and higher frequency control algorithms, in the range of 100KHz to 2MHz PWM, and usages of high frequency switching GaN/SiC power transistors for battery charger, DC/DC converter and inverter control applications. Traditional multi-core microcontroller-based electronics are facing software bottleneck when dealing with higher computation speed and fastest real-time actuation. Increasing operating frequency or adding more cores is a dead-end due to heat dissipation limitation and software design complexity.

This paper introduces a new kind of semiconductor solution, called Field Programmable Control Unit (FPCU), combining software and hardware flexibility, associated with a seamless development and calibration framework tuned for hybrid and electric powertrain control. The FPCU removes software bottleneck by enabling software and hardware execution of high frequency and precise PWM control algorithm while limiting heat dissipation. The associated framework enables the adequate hardware/software split to solve the software bottleneck and smooth parallel software / hardware design, ensuring entire development coherency along the V cycle, from model in the loop (MiL) to hardware in the loop (HiL) design, validation and calibration. The paper presents examples of the improvement reached by this new solution on battery charger, DC/DC converter and inverter application with a gain up to x20 on the control loop frequency in regard to current solutions.

Keywords: HEV, EV, powertrain, inverter, converter

1 Introduction

Improving the performance of hybrid and electric powertrain is directly related to power efficiency increase, battery range extension and battery charging time reduction. OEMs and Tiers 1 are already playing on several factors and investigating / testing new technologies or ideas to find the best solution: battery technology (Lithium-ion, Solid state, Aluminum-ion, etc), electric motor design (3, 6 or 8 phases Permanent Magnet Synchronous Motor), electric motor voltages (24V, 48V, 700V, etc), electric motor

positioning in the car (front belt, gearbox, transmission, axle, wheel), power components (MOSFET, GaN, SiC, etc).

No adequate solution has been offered by traditional semiconductor vendors to control these new systems efficiently. Tiers 1 and OEMs are forced to reuse microcontrollers originally designed for internal combustion engine control and are facing software processing bottleneck with such architecture.

Among the different factors that directly affect the performance and efficiency of a DC/DC, an AC/DC charger or an inverter, the control method (power transistor architecture and technology), the switching frequency (real-time control, PWM frequency and transistors switching) and the digital processing and control method (algorithm executed into the digital controller) are the major ones.

With the emerging of GaN/SiC transistors, complex control methods with higher switching frequency, higher than hundreds of KHz for a Field oriented Control of an inverter and up to several MHz in some DC/DC or AC/DC applications, are required to improve energy efficiency conversion, system size and heat. The switching time is then required to be in the range or below 1 µs.

Additionally, the sophistication of these control commands force the adoption of model based design for the code generation and calibration.

The paper presents the sources of the current limitations of conventional microcontrollers' architecture and introduces brand new semiconductor architecture, Field Programmable Control Unit (FPCU), designed for automotive powertrain control application and its development framework breaking the actual limitations. Some examples of FPCU usage into a Inverter, AC/DC charger and DC/DC converter applications illustrates the resulting benefits of such approach.

2 Software bottleneck issue with conventional solutions

Current solutions face huge limitations making it impossible to reach high frequency with conventional digital control processors. Traditional semiconductor solutions are microcontrollers (MCUs) which rely on multiple proprietary CPU and integrate complex timer-based micro-CPU to handle real-time control of actuators and sensors. These conventional architectures result in a sequential and centralized data processing. They are adapted to complex system control where the maximum response time is within the range of a few milliseconds as is required for internal combustion engines. Needless to say, using products built to sustain a response time of 1ms to control an engine requiring a switching time of 1µs is a mismatch!

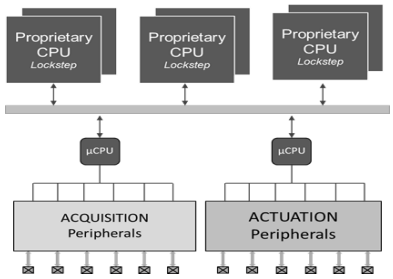


Figure 1: Conventional MCU architecture

For the next generation of electrification control systems, these microcontrollers face software bottleneck issues. A multicore MCU dedicating one of its CPU to the execution of the control loop may barely achieve a PWM inverter control frequency in the range of 10 to 20 KHz in the best case scenario. In the defence of these MCU devices, they are simply not done to execute control loop in the range of 1µs, because there are limited by their inherent sequential data processing capabilities of theirs integrated DSP or FPU that just cannot deliver the sufficient data throughput when the systems requires more than 16 Mbytes/s for the acquisition processing.

MCUs rely only on embedded CPU or fast timers which are sequential processing based. Actual CPU loads directly influence processing time of events, requiring strong engineering efforts to fine-tune timings with a

real-time O/S. Fixed response times whatever the occurrence of the event or the number of events to process is impossible to demonstrate in such architectures.

MCUs need to run very fast to execute the control loop at its fastest speed. Advanced MCUs will run at 350MHz with a fully loaded CPU, resulting in maximum power consumption of the chip and maximum heat generation which directly leads to another big issue for heat sensitive systems like embedded inverters attached to electric motors.

3 Solving the design issues

To solve the control command design issue for high operating frequency the Field Programmable Control Unit (FPCU) and seamless model-based development and calibration framework shall be considered.

4 Removing bottleneck software with FPCU

A newly approach is proposed in complete rupture with conventional systems: combining a flexible and parallel hardware architecture for the real-time processing and control of sensors and actuators, coupled to a standard CPU. All this is surrounded and complemented with an integrated ASIL-D-functional safety architecture to form a single semiconductor.

The flexible and parallel architecture relies on tightly coupled autonomous actuation/acquisition peripherals and a programmable logic fabric interconnected with parallel data paths. The programmable logic has been designed accordingly to automotive constraints, safety requirements and HEV/EV applications requirements.

The resulting architecture is far more powerful and flexible than a conventional safety MCU and much more adequate than a generic FPGA or any other kind of hybrid CPU/FPGA architecture with respect to the targeted applications. Silicon Mobility has defined this new architecture as a Field Programmable Control Unit, or FPCU.

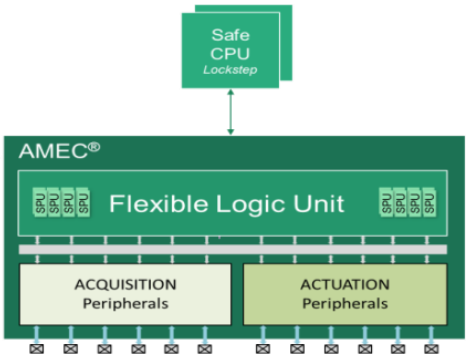


Figure 2: FPCU architecture

A first production release of FPCU is Silicon Mobility's semiconductor solution: OLEA® T222. In OLEA®: the programmable logic fabric is named Flexible Logic Unit (FLU). The FLU includes DSP resources (SPU: Signal Processing Unit) and is linked to generic and registers-based configurable peripherals, so called Powertrain-ready-Peripherals (PrP) to form a dedicated interface for actuators/sensors control and local data processing. This interface is named AMEC® for Advanced Motor Event Control.

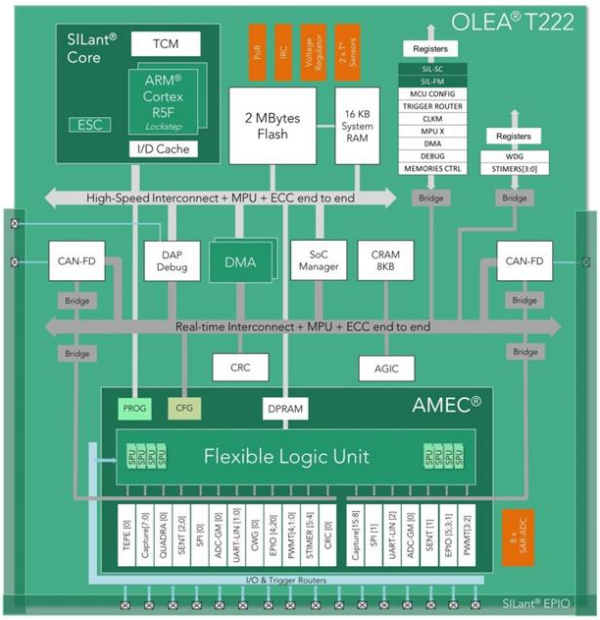


Figure 3: OLEA® T222 FPCU block diagram

With FPCU architectures, the programmable logic fabric hosts the hard real-time processing algorithms and locally processes incoming data from sensors to update actuators without going back to the CPU. CPU is off-loaded and can be used for slower response time functions, releasing it from the timing tuning challenge. In the case of OLEA® T222, the AMEC® is able to process in parallel all data coming from sensors in a fixed response time whatever the number of events to process or their occurrence. The architecture is fully deterministic.

The following figure shows a detailed view of AMEC® inside OLEA® T222. The Flexible Logic Unit integrates several Signal Processing Units (24-bit MAC) which multiply computation power on complex algorithm. (To learn more about AMEC® features and capability, please contact Silicon Mobility).

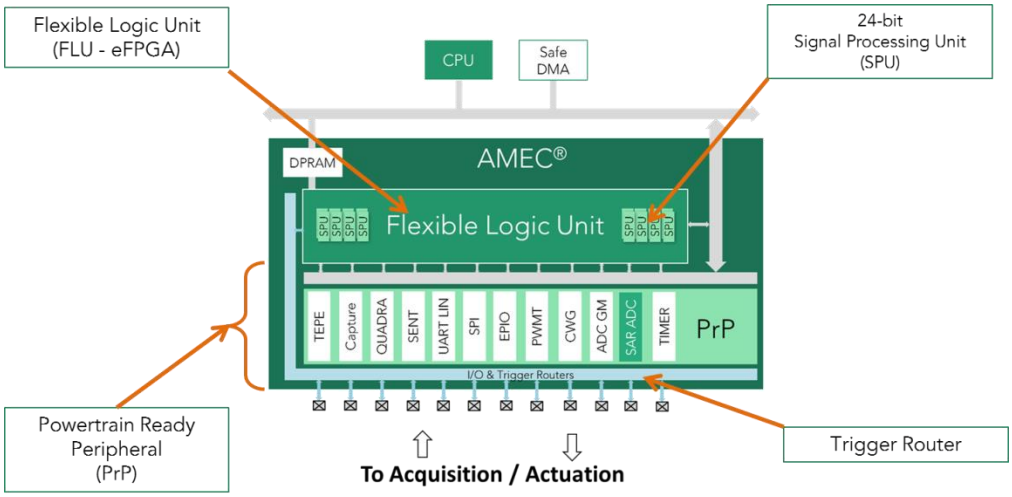


Figure 4: AMEC® block diagram

The Trigger Router is used to route in parallel all inputs (from the PrP, EPIOs and the FLU) to any of its outputs either linked to the PrP, the FLU inputs or towards off chip devices.

With OLEA® T222 FPCU architecture, software bottlenecks are removed. OEM's test-benches have demonstrated an acceleration of 40x on data processing compared to a reference automotive MCU.

Complex control loops as a Field Oriented Control for PMSM electric motor are accelerated by a factor of 20x.

A direct consequence of executing complex algorithms in hardware instead of software is the reduction of the device's power consumption. OLEA T222 has been measured as consuming 180x less than the automotive reference MCU. The FLU can achieve more processing and real-time performances with an operating frequency in the range of some MHz than a processor running above 300MHz.

The FPCU architecture is fully deterministic and integrates functional safety mechanisms, without any processor in the loop, that cover fault at system level. As a result, the Fault Reaction Time (FRT) to address hazard that might arise for inverter, DC/DC or AC/DC control is a matter of nanoseconds instead of several hundreds of µs for solution based on conventional semiconductor solutions.

4.1 Seamless FPCU development flow

To enable a seamless design with FPCU, an entire development framework has been created which orchestrates standard tools all along the V-Model design cycle in place in automotive industry. This framework is named OLEA COMPOSER.

By its unique architecture, the FPCU provides two processing resources (CPU and Programmable Logic - FLU) and several control resources (Peripheral Interfaces - PrP).

The FPCU requires the adequate development framework to enables the adequate Hardware/Software split from the very beginning of the development cycle. When looking to the state-of-the art of the automotive industry development methods: the models are commonly used at least to specify the systems and the main algorithms. Based on these statements OLEA COMPOSER has been defined to both enable the setting of the Hardware/Software split from the very-start of the development cycle and to reduce the development, validation and calibration time and improve performances with an automated code generation providing a comprehensive and high-quality embedded software.

To achieve those goals OLEA COMPOSER plays with the Hardware/Software split using FPCU accurate Target Models (OLEA LIB- Target Models) in the framework from Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) down to Hardware-in-the-Loop (HiL).

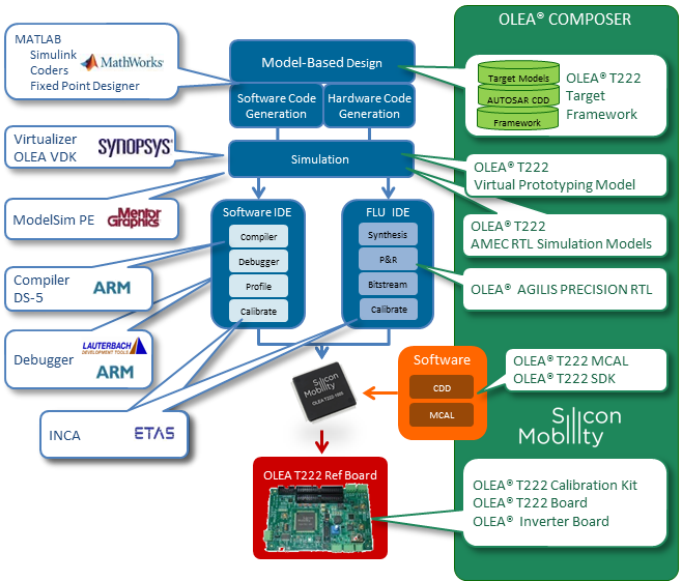


Figure 5: FPCU development framework

4.1.1 Model in the loop

The first stage of the framework starts with a dedicated plug-in for MATLAB and Simulink associated with a library including targets models of all FPCU hardware resources. This set allows from a reference control command model accurate simulation with respect of the FPCU hardware capabilities.

The plug-in implements an automatic design flow where designers are unloaded from handling intercommunication between CPU and FLU or FLU and PrP. Data exchanges, timing and synchronisations are automatically handled by the flow within MATLAB. Designers can work directly from Models and split their algorithm to allocate the right parts to the adequate resources between CPU / FLU / PrP using target models. From a designer stand point of view, the steps are:

- Design its reference control model
- Port its design reference model to FPCU using the target library available
- Select automatic generation of data and timing handlers for CPU / AMEC FLU / PrP communication
- Select variables to probe and parameter to adjust and automatically generate data logger and updater to be used in HiL simulation or calibration phases
- Automatic code generation of .C and .HDL (Verilog or VHDL) from MATLAB Coders
- Automatic compilation of .C and .HDL using standard 3rd party tools.

The generated code out of the framework is structured as an AUTOSAR compliant Complex Device Driver (CDD) which can be directly integrated into AUTOSAR application software. From software designer point of view,

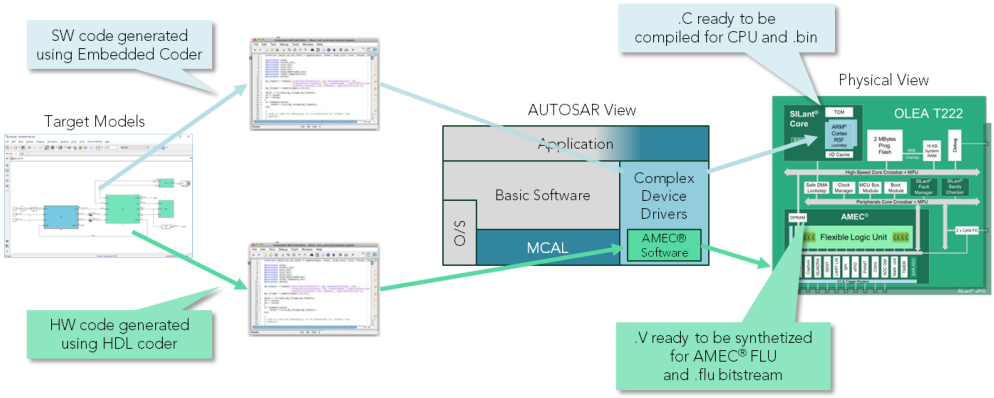


Figure 6: FPCU development framework

As a result the command law H/W and S/W design split could be adjusted from the very start of the development cycle. Codes for CPU and AMEC FLU are automatically generated from MATLAB in a coherent and simultaneous way. This model-based approach enables the design of sophisticated and precise control command.

4.1.2 Software in the loop

Many automotive OEMs and Tiers1 are also moving toward software simulation in their early stage of project development. Software simulation enables fast application development and increase of productivity by using virtual prototypes instead of a hardware target. A virtual prototype of the FPCU has been developed in System C and can simulate the execution of a full AUTOSAR software stack, including the Complex Device Driver previously generated. Building entire Software in the Loop (SiL) stage is made possible with this virtual prototype model. This enables the complete validation of the generated code

In complement to this high level model, a precise and accurate model of the AMEC® FLU has been also developed for Register Transfer Level (RTL) simulation of the HDL generated code. This particular model enables cycle accurate simulation and deep signalling synchronization tuning.

To maximize visibility and validation of the system, Virtual prototype and RTL models can also be used simultaneously in co-simulation.

4.1.3 Hardware in the loop

OLEA® T222 FPCU silicon and Emulation Device development material combined OLEA® T222 Starter Kit allows users to validate their design in HiL environment. The framework is instrumented for measurement and calibration. It leverages the OLEA® T222 programmable resources: variables measurements and parameters settings are non-intrusive, fully observable, modifiable, and independent of their location (CPU or AMEC® FLU) and use a single JTAG or TRACE port interface.

4.2 FPCU proof of concept results in HEV/EV applications

On-going developments and study assesses the actual performances of an entry range FPCU, the OLEA® the T222, jointly with its OLEA® composer and OLEA® LIB for Inverter, Battery charger and DC/DC converter applications.

4.2.1 FPCU usage for inverter applications

The FPCU has been used for inverter control for a 3 phases PMSM electric motor. A Field Oriented Control (FOC) regulation has been implemented first on Matlab Simulink to target an FPCU using the OLEA® LIB Target and OLEA® LIB MATH. The figure below shows the top level view of the complete FOC. The speed regulation which do not require a high operating frequency (below 10 KHz) to calculate the torque reference is implemented into the processor (CPU). The complete FOC regulation from 3 phase currents and motor position to the 3 PWM command calculation is performed by the AMEC® FLU.

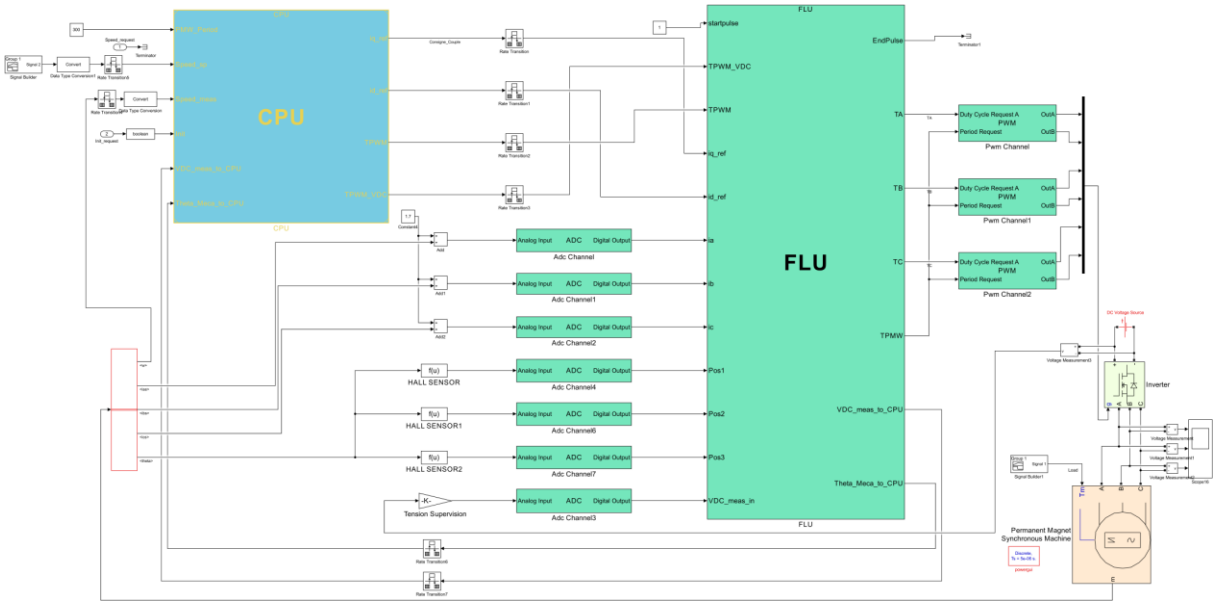


Figure 7: 3 phases PMSM Inverter under Simulink with OLEA® COMPOSER – FPCU Target models

As depicted in the Matlab Simulink model figure above the signals acquisition for the phase currents are relying on the FPCU AMEC ADC (Analog to Digital Convertor). The motor position estimation relies on an ADC acquisition and a tracking loop algorithm running on the AMEC FLU as well. The PWM signal generations rely on the AMEC PWM peripherals.

The figure below depicts the complete FOC implemented into the AMEC FLU. The FOC calculates the PWM duty cycle commands from the 3 phase current ADC acquisitions, achieved in parallel, with Clarke

current transform, Park Current transform, Id/Iq regulation based on the torque commands (Idref/Iqref), Decoupling & Flux weakening, Park Voltage transform and finally Space Vector Modulation (SVM). The AMEC FLU also hosts the Speed/Position tracking loop processing of the Hall-effect magnetic resistive signals to calculate the speed and the position.

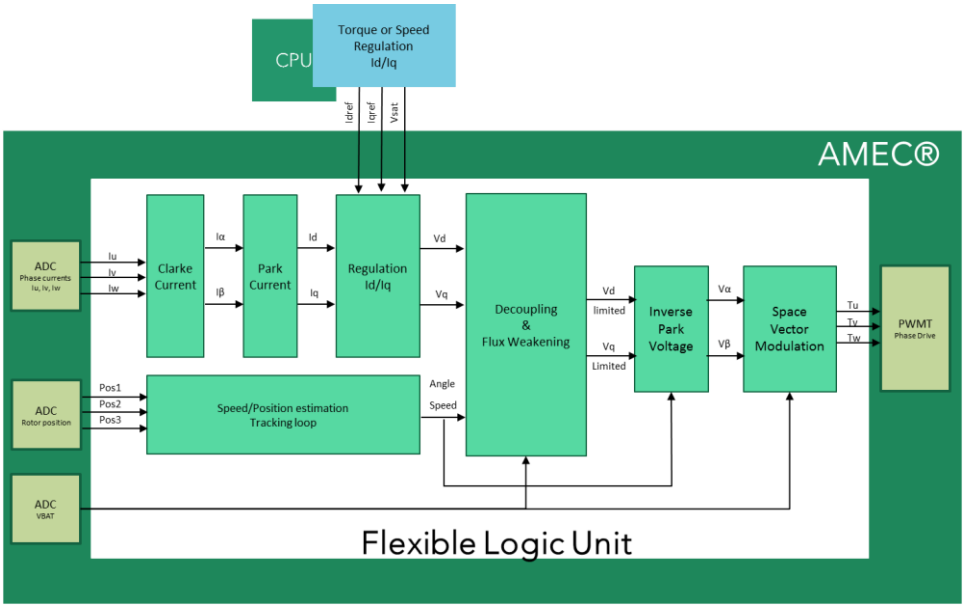


Figure 8: AMEC® FLU content for the 3 phases PMSM control

The whole embedded software to program the AMEC FLU and the processor, including the codes to control the ADC, PWM and AMEC FLU – CPU has been generated automatically with OLEA® Composer -Target framework and the OLEA® LIB T222 Target from the Matlab Simulink models. With this development flow the entry range OLEA® FPCU achieves a 220 KHz control loop operating frequency to generate the PWM duty cycle commands with an AMEC FLU operating frequency of 10 MHz The processor load is less than 2% with an operating frequency below 70MHz (The T222 FPCU maximum CPU operating frequency is 200MHz). As a result a very low current consumption and a very limited heating are anticipated.

The figure below shows the hardware material demonstrating these performances. This demonstrator is based on:

1. On OLEA® T222 FPCU evaluation board
2. Inverter board
3. PMSM 3 phases 24 volts
4. Motor load

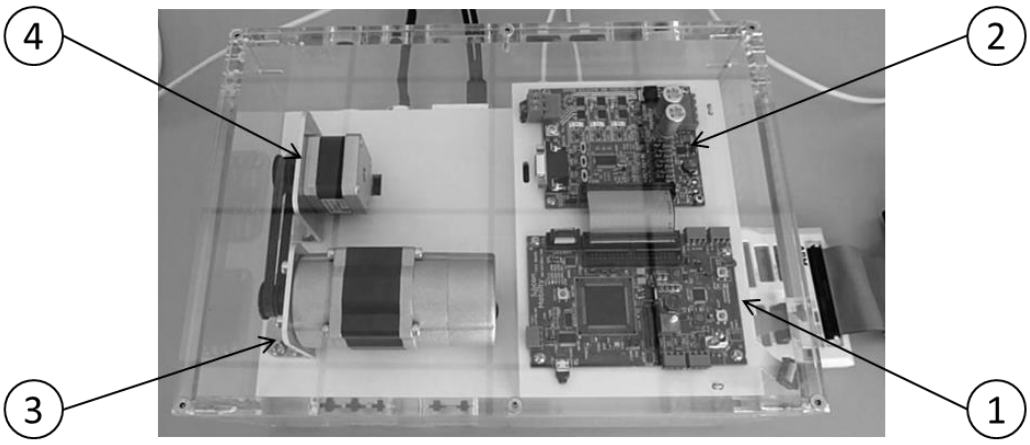


Figure 9: OLEA® T222 FPCU 3 Phases PMSM control demonstrator

The faster the regulation is running the more the energy of the harmonic currents, which are in particular at the origin of the electric motor over-heating, will be limited. This will be especially backed by the GAN/SIC transistors technologies which limit the energy loss on high-frequency switching.

The FPCU also reduces the Bill-Of-Material (BOM) especially for off-chip diagnostic in discrete logic or for off-chip regulation close to the high-power stages. The FPCU has been also used for a 6 phases WRSM (Wounded Rotor Synchronous Machines) motor. For this application the FPCU demonstrates the economy of 72 off-chip components in comparison with a conventional automotive dual-core microcontroller.

The BOM reduction could be noticed in particular for over-current detection. As such for conventional MCU, the measurements of phase currents of the electric motor is carried out by a Hall-effect sensor which converts the phase current of the machine to a voltage value provided to the microcontroller analog converter input. This sensor voltage is also provided to 2 off-chip comparators which compare the output voltage of the Hall-effect sensor to two reference voltages for both positive and negative over current detection. An over-current is detected when the output voltage of the Hall-effect sensor is below a defined low level voltage value or above a defined high level value. The comparator outputs are connected to a microcontroller digital input to signal the status of the over-current.

In return the entry range OLEA® FPCU T222 embeds 2 ADC modules that incorporate different types of comparators to optimize the BOM:

- A hardware comparator with two external reference inputs in each ADC cell
- A logical hardware range checker (software configurable) for each external analog input

This allows the FPCU to detect over-current fault in the system with a minimum of additional electrical components such as external comparators and their associated passive components.

In addition the FPCU offers a comprehensive integrated function safety architecture, as depicted in the figure below, that has been also implemented for a 6 phases WRSM electric motor to detect and contain application faults without any processor intervention.

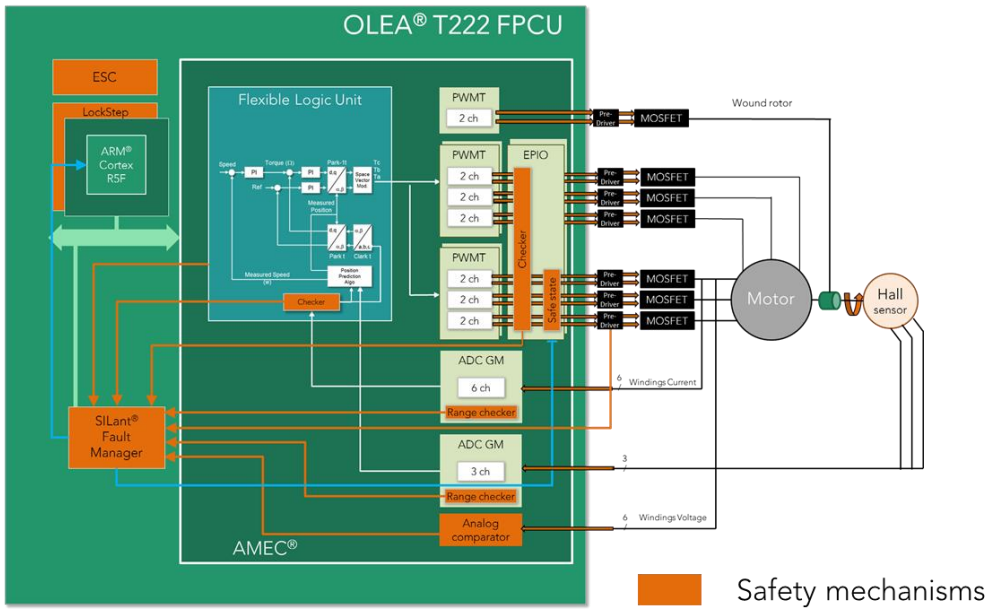


Figure 10: OLEA® T222 FPCU – Safety Mechanisms for 6-phase WSRM

The FPCU integrates the following safety mechanisms optimized for such applications:

- Processor in lockstep
- PWM and EPIO in lockstep and associated checkers
- EPIO safe state mechanism applying the safe levels in case of fault
- Analog comparators for over-voltage, over-current and short-circuit
- Range checker for the phase currents and Hall-effect voltage measurement
- Event Sequence Checker to verify the correct hardware and software events sequencing

- Application dependent safety mechanism that could be hosted in the AMEC® FLU
- The Fault collector and Manager (SILant® Fault Manager) that acknowledge the fault and contain the fault in accordance

This type of integrated functional safety enables Fault Reaction Time (FRT: time between the fault occurrence and the fault detection), without any CPU intervention thanks to the AMEC® FLU and safety mechanisms, duration below 370 ns whereas the conventional MCU cover it in several milliseconds or requires off-chip components. This has been measured for over-current, over-voltage, short-circuit, sensor-loss and phase loss.

4.2.2 FPCU usage for AC/DC charger applications

The FPCU has been used for an AC/DC converter: from a 3 phase sources voltage (400 Volts phase to phase) to supply 400 volts batteries.

As depicted in the Matlab Simulink model figures below, the signals acquisition and processing for the voltages and currents are relying on the FPCU AMEC ADC running at 2 MS/s for 7 conversions in parallel and a decimation filter hosted in the FPCU AMEC FLU. The whole regulation to calculate the 6 PWM duty cycle commands to control the IGBT according to the 3-phase electric power sinusoids is processed in the AMEC FLU. The PWM signal generations rely on the AMEC PWM peripherals. The faster the regulation is running the more the IGBT commutations are generated in the appropriate time slot to favour the optimal yield and reduce the ripples. This reduces the current source harmonics with unity power factor.

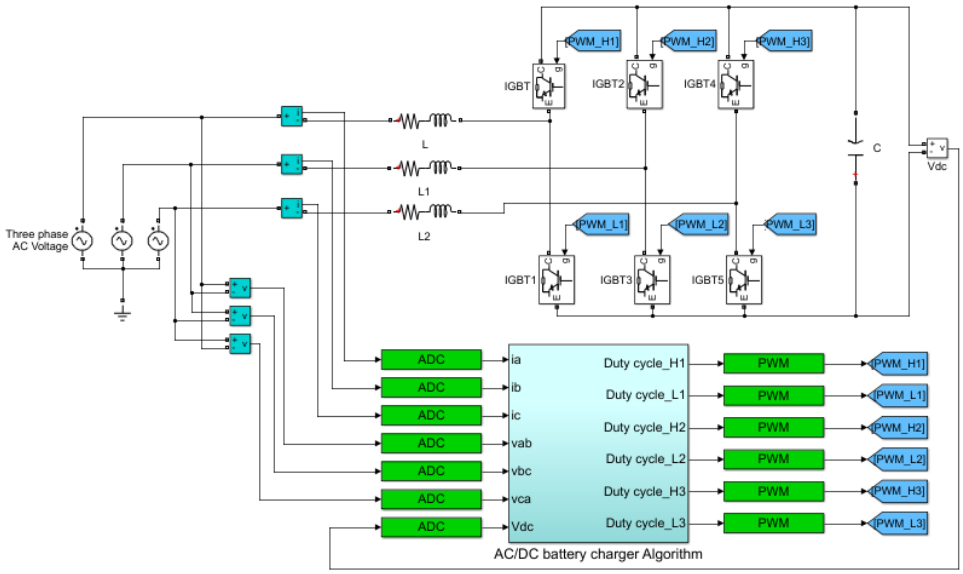


Figure 11: AC/DC charger – Matlab Simulink view

The complete embedded software to program the AMEC FLU and the processor, including the codes to control the ADC, PWM and AMEC FLU – CPU have been generated automatically with OLEA® Composer Target framework and the OLEA® LIB target from the Matlab Simulink models depicted below. With these automated code generation system based on the FPCU achieves a 200 KHz control loop whereas the conventional dual-core microcontrollers hardly achieve a 100 KHz based on the regulation coded manually.

4.2.3 FPCU usage for DC/DC converter applications

The FPCU semiconductor has been used in a Bidirectional DC/DC converter Buck-Boost electric implementation to host a “Valley Current Control” command law.

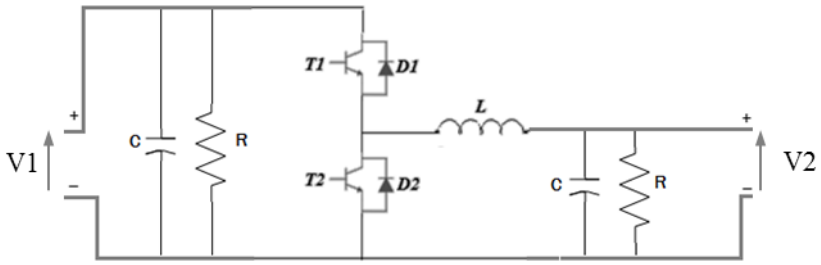


Figure 12: Bidirectional DC/DC converter Buck-Boost electric scheme

The Valley current control law accounts among the best DC/DC converter regulations in terms of energy yield, ripple control, stability, robustness, power range and response time. The table below ranks the different type of digital command laws.

Table 1: DC/DC Converter command law

Command Laws	**Yield**	**Ripple Range**	**Stability Robustness**	**Over Current/voltage**	**Response Time**	**Power Range**
Valley Current Control	98%	Limited	High	Limited	Very short	2-10 KW
Average Current Control	97%	Limited	Low stability	Limited	Short	2-10 KW
Delayed Peak Current Control	97%	Limited	Low stability	Limited	Long	2-10 KW
Prediction Current-Mode Control With Delay Compensation	97%	Limited	Low robustness	Limited	Long	2-10 KW
Compensated Digital Current-Mode Control	97%	Limited	Low robustness	Limited	Very short	2-10 KW
State Feedback Current Control	94%	Limited	High	Limited	Very short	2-10 KW
Average Voltage Control	96%	High	Medium	High	Very long	< 2KW
COT DPV	96%	High	Medium	High over-current	Long	< 2KW
PWM DPV	96%	High	Medium	High over-current	Long	< 2KW

The figure below presents the DC/DC converter regulation implemented with the FPCU. The ADC and PWM are part of the AMEC PrP. The Voltages controller, Delay and Valley Current Control functions are hosted by the AMEC FLU.

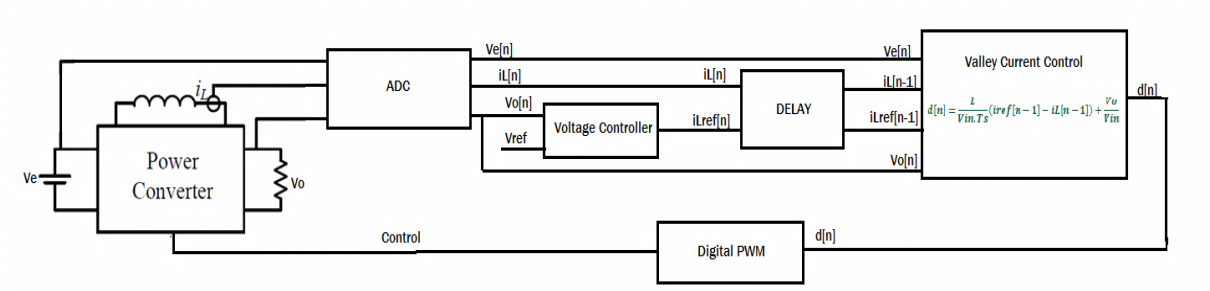


Figure 13: DC/DC converter Valley Current Control scheme

The Valley Current Control implemented on the FPCU from Matlab Simulink as shown in the figures bellow achieves a 200 KHz control loop frequency to calculate the new PWM commands for 6 DC/DC converters in parallel with less 2% of processor load.

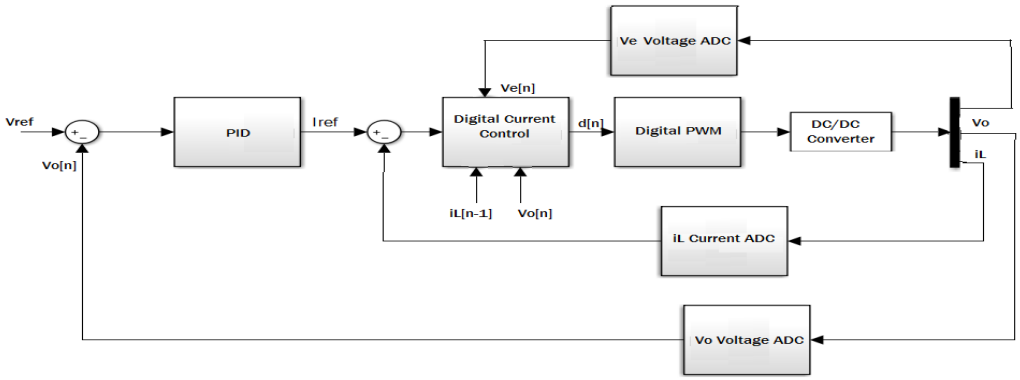


Figure 14: DC/DC converter Valley Current Control Simulink view

These results have been compared with the same Valley Current Control command law running on an automotive dual-core microcontroller. The Valley current control loop based on the FPCU runs 6 times faster.

The Valley current control loop frequency has direct impact on the energy yield and on the ripples. The ripple energy is inverse proportional of the control loop operating frequency. Increasing the frequency improvess the quality of the DC/DC converter.

Table 2: Ripples range formula

Ripples range	
Current control loop	$\Delta i = \frac{Ve.\alpha}{L.F}$
Voltage control loop	$\Delta Vo = \frac{Ich.\alpha}{C.F}$

Indeed the FPCU integrated functional safety architecture provides the adequate mechanism to detect and contain an overvoltage or an overcurrent hazard, without any processor intervention, with a fault reaction time below 100ns. This is supported in particular by the integrated analog comparators.

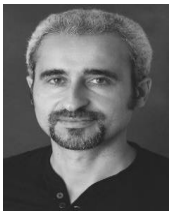
References

[1] Khaled Douzane - *FPCU – New Semiconductor Architecture to Boost HEV & EV Systems*, http://www.silicon-mobility.com/download/, 2017©

[2] Silicon Mobility, *OLEA T222 Product brief*, http://www.silicon-mobility.com/download/, 2017©

Authors

Anselme Lebrun has 22 years of experiences in the semiconductor industry. He is Advanced Technology Director at Silicon Mobility. He is currently driving innovative technologies for powertrain and autonomous vehicle applications. Anselm Lebrun is graduated from *ENSEIRB* engineering school in electronics.



Khaled Douzane has 17 years of experiences in the semiconductor industry. He is VP Product at Silicon Mobility. He is currently driving product lines for HEV/EV powertrain and autonomous vehicles applications. Khaled Douzane is graduated from Nice Sophia-Antipolis POLYTECH engineering school in electronics