International
IOR Rectifier

New Digital Hardware Control Method for High Performance AC Servo Motor Drive – Accelerator™ Servo Drive Development Platform for Military Application

Toshio Takahashi, International Rectifier

As presented at Military Electronics Conference, Sept 24-25, 2002

Abstract:
Today's motor drives widely use Digital Signal Processor (DSP) or Microcontroller to implement the digital control algorithm. Most recently new requirements have arisen. These include faster torque control update with flexible design capability of motion peripherals for high performance military servo drive applications.
A Complete digital hardware based AC servo drive development system has been developed to satisfy increasing demand for performance enhancement. Based on the FPGA, the system is configurable for either induction or permanent magnet machine servo control. The detail design of complete hardware based high performance AC servo drive system is discussed.

Introduction:
Today's digital AC drive consists of implementation of torque control by means of regulating the motor current which requires very high speed computation in the range of tens of microseconds, and speed control which requires relatively moderate computation intensity in the range of hundreds microseconds. All these functions have been implemented in one DSP or one microcontroller with/without a separate motion ASIC which performs much faster tasks such as the PWM (Pulse Width Modulated waveform generator), encoder signal interface, etc. This popularity has been due partly to availability and flexibility of desired algorithm implementation.
However, the performance envelope is further expanded and application is further diversified, new issues arise to satisfy the need for new performance challenges.

1) Inadequate performance due to lack of computation power for high-end servo torque control resulting in multiple DSP or microcontroller implementation mixed with specific motion peripheral ASIC(s);One for torque control and the other for other controls. This creates additional complexity associated with functional partitioning and interconnectivity among split modules/functions.
2) Impeded process and challenge to meet faster time-to-market development. Exponential effort in programming is already inevitable in today's drive development process. A Specific coding technique is often used to meet demanding challenges, and to satisfy new computational speed requirements. Code maintenance ability is further deteriorated if a cryptic assembly language is used as opposed to a high level language such as C in order to meet the speed requirement. In reality, the torque control algorithm is particularly written in the dedicated assembly language, which adversely affects programming productivity.
3) Lack of hardwired algorithm solution and determinism. Any DSP or microcontroller uses memory devices to store instructions. However, this may create a single bit event upset for military applications. Recent DSP and microcontroller heavily incorporate deep pipeline and cache memory for their execution units. When, for example, a branch instruction is executed, these pipelines are flushed and reloaded with new fore-executing instructions. Therefore, it is inherently impossible to maintain consistent overall execution time. Fluctuation of computation speed of current control loop and PWM unit is sometime catastrophic and raises reliability issues.

In order to solve the arising problems stemmed from a traditional DSP and microcontroller approach, the new method of hardware control has been developed. More than ten times faster execution of current control has been achieved when compared with a traditional DSP. New algorithm development method is also proposed. Unlike traditional programming/coding effort, it simplifies the method of algorithm development by introducing high-level graphical design by

Control Blocks. This paper reviews today's digital servo drive status and issues, and introduces new solution for military application in order to satisfy demanding servo drive performance with Accelerator™ servo drive development system.

Review of the today's servo drive - digital control architecture and functional partitioning

Today's most servo motor drive systems are implemented by digital closed loop control instead of analog control. This has been due primarily to rapid advancement of Digital Signal Processor (DSP) and microcontrollers applied to motor control applications. In a typical servo control system, several functions are divided into tasks which runs at different update rates depending on the required bandwidth and nature of the processing priority need – real-time operation versus delayed batch processes, scanned tasks versus one-time event driven tasks. Each task is controlled by a multitask operating system closely coupled with DSP or microcontroller interrupt structure.
Figure 1 shows a typical structure of servo drive system in terms of function. Functional element, which deals with much closer machine control, requires fast processing, fast update rate and real-time process. They are closely tied with a specific motion peripheral hardware and it sometimes requires specific coding unique to peripheral hardware and interrupt structure inside of DSP or microcontroller.

On the contrary, tasks that are far apart from the machine side and are close to the host communication or man-machine interface side, require less frequent update and slow processing. However, it requires more memory intensive calculation since reference command generation over controlling parameter is more complicated than those, which are close to the machine side. For example, position reference command is much more complex as sophisticated motion profile generation advances. However, torque command is produced in a simple step function.

The fact is that torque is a fastest machine parameter, and needs to be controlled much more quickly than speed of motor shaft. Integral of torque is speed. Integral of speed is position. Integral of power results in motor temperature rise. Because of this chain of physical motor parameters, each parameter requires different speed of processing. It is typical that a real-time multi-tasking operating system is used to satisfy each required processing power.

However, today's DSP or microcontroller are facing more difficult and fundamental challenge to deal with all these tasks requiring different processing power.

Today's servo drive requires different effort and enhancement in two diversified directions. Controlling torque requires much faster computation update rate to sustain the closed loop bandwidth and increasing demand of dynamics of the machine. For example, in order to achieve 4kHz torque bandwidth, torque control loop computation update rate requires 25 or less microseconds computation time range.
The role of power management function increases. More advanced deadtime compensation, newly arising EMI noise reduction associated with hard switching, close monitoring of temperature rise of power devices, accurate motor phase voltage feedback sensing, current sensing, and motor machine parameter estimates are emerging functions which requires more specific programming, fast computation, unique digital hardware logic and analog circuit. Processing speed dealing with these logic reaches nanoseconds order and usually implemented by the dedicated ASIC.

On the other hand, task dealing with position control and host communication require more memories and programming effort. More sophisticated motion profiling requires more memories in both instruction and data with complex programming and decent computation update rate.

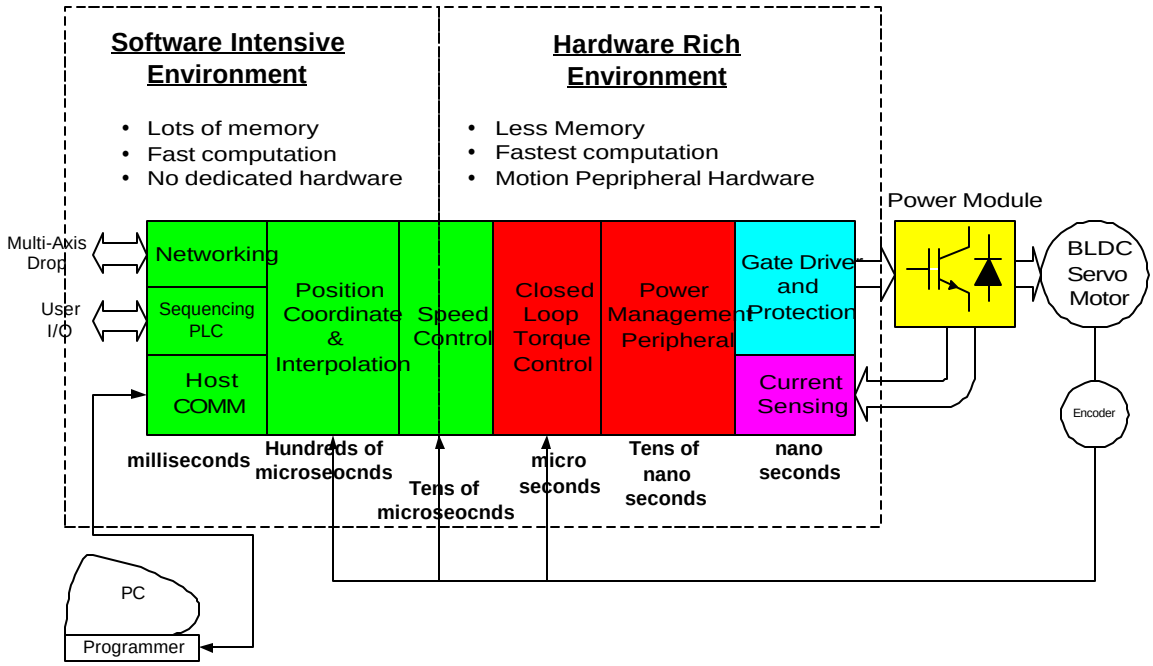

Figure 1 Servo Motor drive functions

As shown in Figure 1, computation requirement is basically divided into two parts. Hardware rich environment requires fast computation and more intensified dedicated hardware logic in power management peripherals. Software intensive environment requires a lot of memories with more programming effort.

It is impossible to achieve both side of function all together in one DSP or microcontroller. Today's mostly available motion-DSP does not adequately satisfy this need. Lack of computation power, exponentially increasing demand of memory for programming and data, and new hardware logic in motion peripheral are difficult to implement without compromise of performance and complexity of gate logic.
If today's motion-DSP tries to satisfy the need without modifying fundamental structure of DSP, the chip becomes impractical and less economical.

It also becomes difficult to implement all of motion functions in ASIC plus DSP in the framework of traditional functional partitioning, where all of closed control including torque control, speed control, and positioning control are implemented in one DSP or microcontroller, while motion peripheral function is implemented in the dedicated ASIC. This is due largely to fact that toque control demands much faster computation update rate thus it becomes difficult to perform toque control function along with other control function within a DSP or microcontroller software computation environment.

Traditional approach is quickly approaching limit not being able to satisfy performance demand, faster development time, and easy and low cost software maintenance.

Accelerator™ servo drive architecture

International Rectifier has recently introduced Accelerator™ system. It is a servo drive development system, which simplifies the development process of digital servo drive system while achieving high bandwidth in torque control response. Development system consists of the development platform and the development tools.

Performance advantage – hardware torque loop
Time critical control execution is implemented in hardware rather than in software. Latency and execution time does not vary. It is inherently fast and deterministic. Complete closed loop current control is implemented by Verilog code with a FPGA (Field Programmable Gate Array). Control block diagram, shown in Figure 2, is

based on the field orientation in synchronously rotating frame. Three-phase motor current are decomposed to the d-q components via vector rotator and Clark transformation, and regulated by two independent Proportional plus Integral amplifiers respectively. The resulting d-q frame voltages are converted to stationary frame coordinates and passed onto Space Vector PWM modulation units.

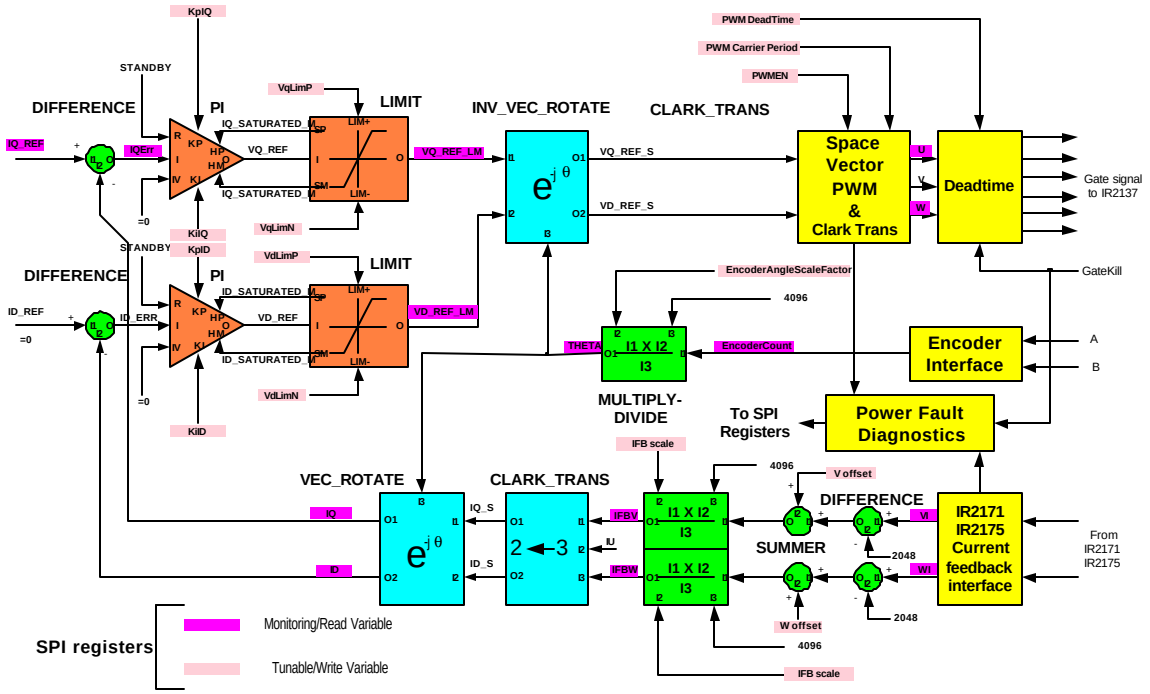

Figure 2 Control Block Diagram

Each block in Figure 2 is implemented in Verilog code to realize hardware logic circuits. For example, the detailed block diagram of Proportional plus Integral is shown in Figure 3. It contains two 16-bit multipliers and two 32-bit accumulators. It also has ability to hold Integrator in the event of output reaching to saturating limits. This block only takes 13 system clocks to compute which is equivalent to 260 nanoseconds at 50MHz system clock. In comparison, it takes 6-7 microseconds to compute by today's most recent 16bit DSP which is written in assembly language.

	System Clock	Computation time @50MHz clock
Proportional plus Integral	13	260 nanoseconds
Vector Rotator	11	220 nanoseconds
Clark Transformation	2	40 nanoseconds
Encoder Interface	2	40 nanoseconds
Space Vector Modulation	11	220 nanoseconds
Multiply and Divide	3	60 nanoseconds
Difference	1	20 nanoseconds
Inverter	1	20 nanoseconds
Ramp	2	40 nanoseconds

Table 1 Each block computation time

Computation time associated with each blocks are listed in Table 1. Excluding the time associated with current feedback discretization (analog to digital conversion process), which is approximately 8 microseconds, the whole current control computation is executed at 1.1 microseconds with a 50MHz system clock. This enables high update rate of current control.

When run at 20kHz PWM carrier frequency with 40kHz computational update rate, torque control loop response reaches 5kHz at –3dB point. This high bandwidth torque control allows interfacing with recent low inductance servomotor such as linear motor in order to achieve less harmonic current ripple.

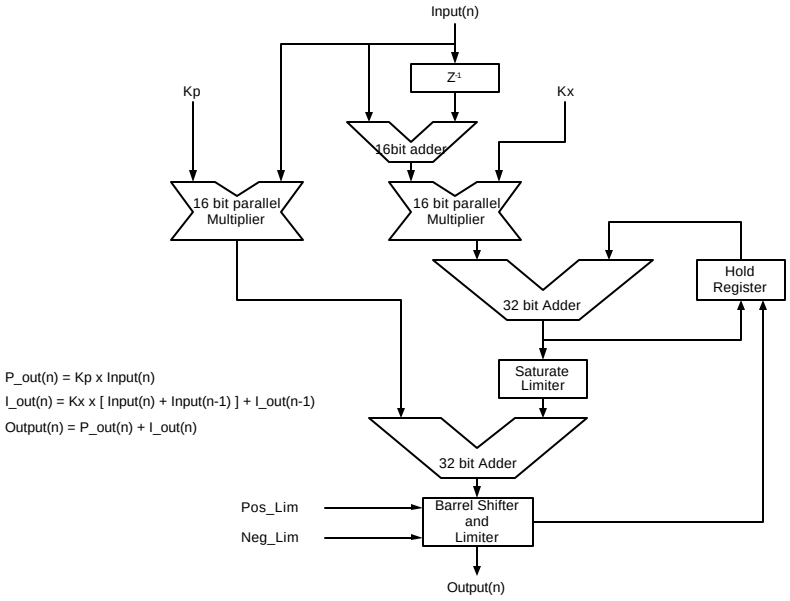

Figure 3 Proportional plus Integral block

Closed loop bandwidth was measured by applying sinusoidal excitation signal to the reference input while the rotor was locked. Figure 4 shows the bandwidth measurement data taken by the HP35665A dynamic signal analyzer. PWM carrier frequency is set to 20kHz and FOC update rate is set to 40kHz. The excitation sine swept signal level is ±200mV, which corresponds ±200 digital count value and correspondingly ±4% of rated current. Input Iq reference sine swept signal is delayed through A/D converter and internal discretization process while output signal is directly sampled by the Tektronix AM503B current amplifier with Tektronix 6312 100MHz current probe. Input time delay at measurement through A/D converter is approximately 12 microseconds, which is equivalent to 22° phase offset at 5kHz.

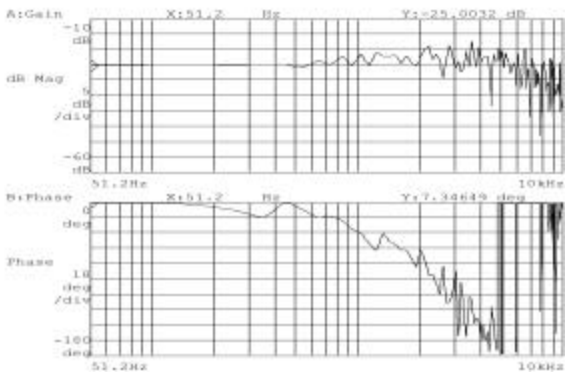

Figure 4 Closed current control Bode Plot

Therefore, the actual phase margin at 5kHz from the graph chart should be read about – 155 degree as opposed to –170 degree in Figure 4. The plot also shows gain magnitude point at -3dB reaches 6kHz.

Accelerator™ hardware construction

Figure 5 shows the overall Accelerator™ servo drive development platform hardware. The system can handle up to 1.5kW output with 300%/1 second overload yielding 230V/7Arms continuous power rating. Position feedback is incremental encoder interface with index marker pulse. Input voltage can be AC115V or AC230V and either single phase or three-phase configuration. The system is equipped with variety of protection circuit ranging from over-current/short circuit protection circuit, over-temperature protection circuit, and over-voltage protection circuit.

The hardware consists of two pieces of boards as shown in Figure 5. The intelligent power board (IRACP101) contains the FPGA with pre-configured hardware torque loop algorithm, IGBT power module, high voltage IC based gate drive (IR2137) and high voltage IC based current sensing (IR2175), encoder interface circuit, and fly-back power supply. Overall Power conversion circuit is highly integrated with latest motor drive chip set from International Rectifier and total circuit is very simple and compact.

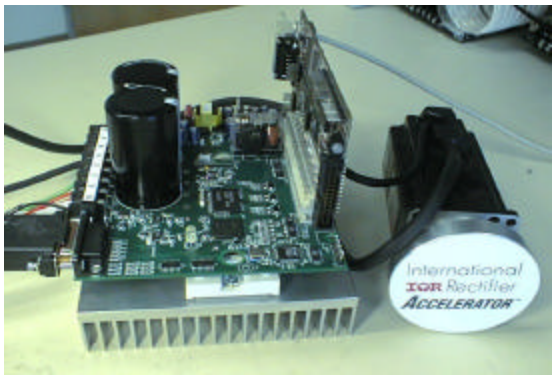

Figure 5 Accelerator™ development platform

The system comes with the complete schematics, the Bill of Materials. User can purchase optional layout database file separately to modify/create own version of PCB.

The IGBT module, rated at 600V/30A, is an industry standard ECONO-PIM2 module and has six IGBT/FREDs at inverter section, input diode rectifier (six diodes), and brake IGBT/FRED. The module also contains a thermister for over-temperature protection. The gate drive is based on the International Rectifier latest 3-phase high voltage gate driver IC, IR2137, which has a built-in over-current protection circuit on the chip. For higher overload rating application such as high performance servo system,

The second board (IRACP201) contains DSP (TMS320C6211), FPGA, Flash memories, SDRAMs, 8 channel A/Ds, and 4 channel D/As. The FPGA on board is pre-configured and contains peripherals such as QSPI (Queued SPI) serial interface module, RS232C module, and optional encoder interface. This board is designed to implement less time critical and more memory intensive tasks such as velocity control, position coordinate, motion profiling, user interface I/Os and communications.

Accelerator™ system Design Flow and Design Tool

Figure 6 shows the basic block diagram of hardware torque control implemented in the FPGA on the BrainPower™ board.

There are two levels of programming and configuration ability in the system.
The first level is mainly configuration without changing control structure by any development tools. All parameter change is initiated through the Graphic User Interface program (GUI) shown in Figure 7. With this GUI, the following parameters and hardware can be configured and changed.

- PWM carrier frequency/mode and Deadtime
- Gains of torque regulators
- Encoder line count and Hall indexing initialization data
- Motor current

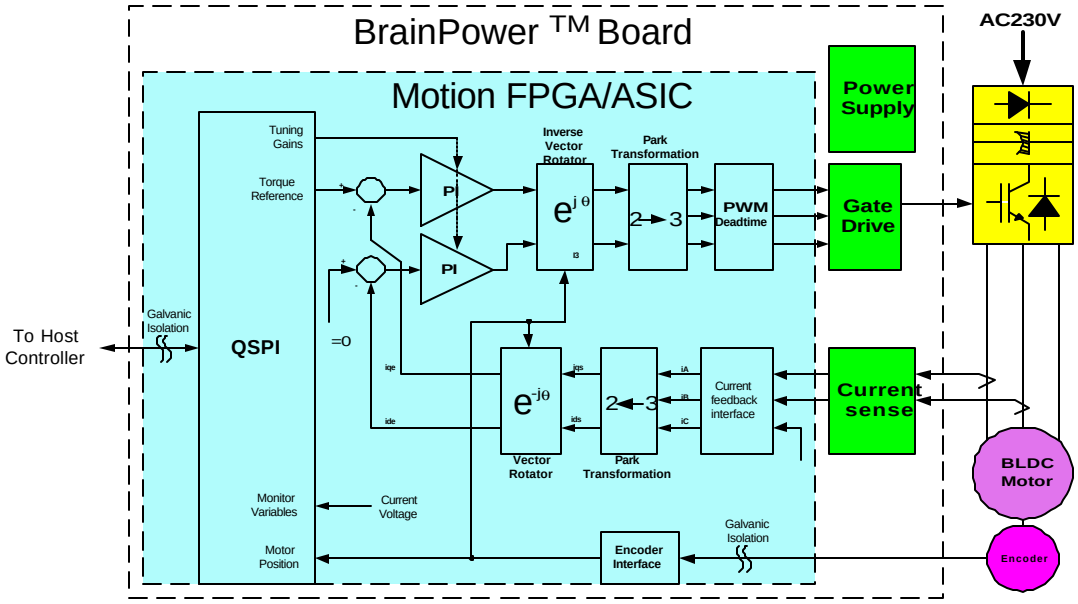

Figure 6 Hardware torque control with IRACP101 Intelligent Power board

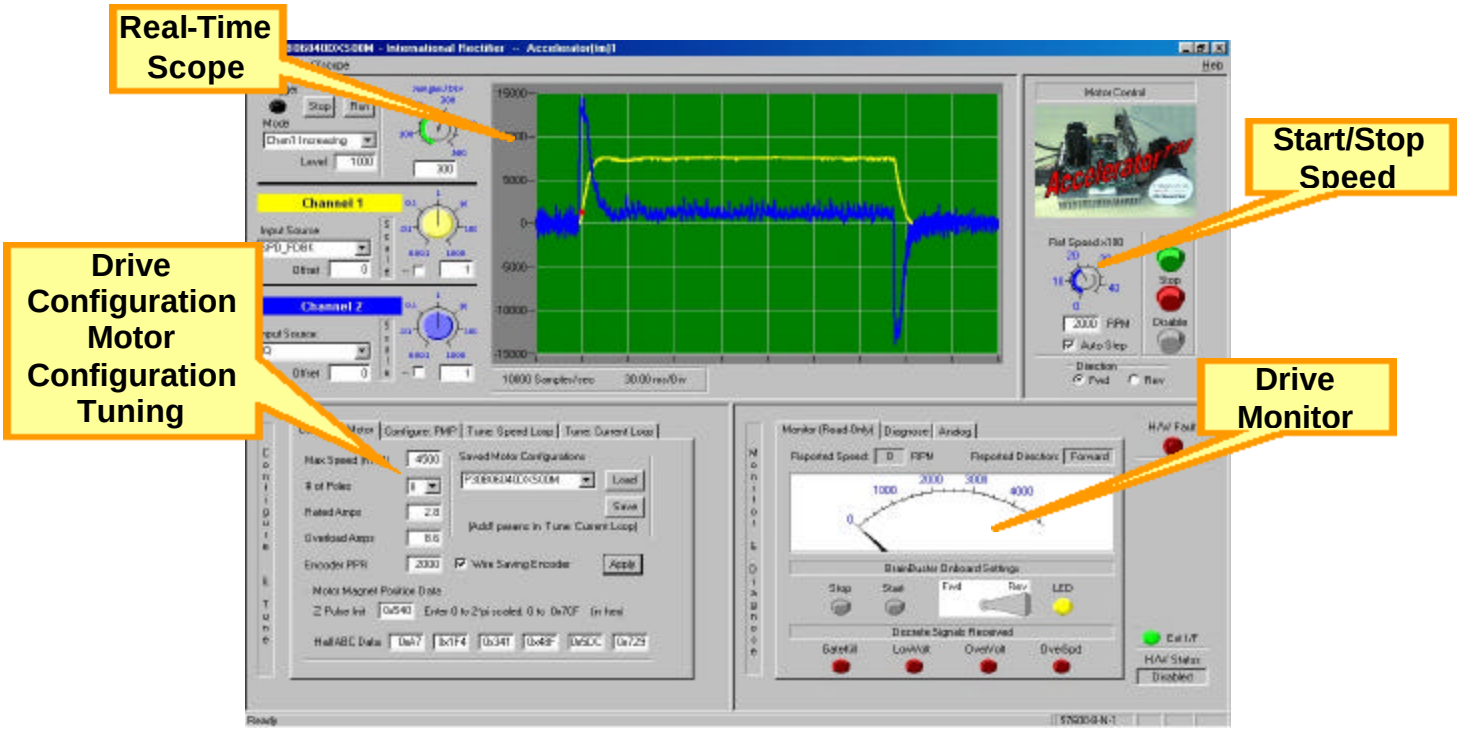

Figure 7 Graphic User Interface Configurator

In addition, the GUI allows real-time monitoring function by two-channel scope on the PC for any signal variables within the hardware current control system.

The second level of programming and configuration involves structure change of the closed loop control such as adding slip gain feed forward blocks for induction motor application, and replacing position feedback interface unit with resolver feedback interface unit.

Programming in the Accelerator™ system is very much different from traditional programming of DSP or microcontroller. As mentioned earlier, traditional programming method usually requires particular skill such as memorizing hundreds of instruction set, addressing mode, register structure. And all of these are uniquely tied with a particular DSP hardware structure inside and require intensive knowledge of the machine architecture itself.

MVP – Matlab™ to Verilog Porter

Accelerator™ programming method is much more focused on functional level of programming. Unlike traditional DSP/microcontroller programming, it does not require specific knowledge of programming and/or unique hardware structure of FPGA device. It basically consists of Matlab™-to-Verilog Porter (MVP) and unique Accelerator™ control block library. The motion control algorithm is described by graphical control block language, and the design is entered through Matlab/Simulink™ environment with Accelerator control block library. Graphical editing of control blocks is very much similar to the block diagram shown in Figure 2. Typical programming is completed just within hours instead of months. Once a design is entered, it will directly compile into Verilog code followed by synthesizing to target FPGA. Final hardware implementation code is then downloaded into the FPGA and the Intelligent Power board (IRACP101) becomes ready for testing with the real motor. Matlab/Simulink™ simulation models are also provided for each control blocks. These simulation models have exact matching to each Verilog code control blocks in terms of data range, data format and resolution. Thus system level simulation can be accurately performed prior to downloading the code to the target FPGA.

Accelerator™ control block library is a collection of primitive control blocks, which are described by Verilog code and fully tested. Difference, Multiply/Divide, Proportional plus Integral, Vector Rotator, Space Vector PWM, QSPI (Queued SPI serial communication interface) are a just few examples. With the MVP tool, a designer selects blocks according to desired motion control function and connects each block by signaling paths to form the final control systems. Unlike traditional DSP or microcontroller, desired motion peripherals such as encoder interface versus resolver interface are configurable and easily customized. Depending on specific need for feedback devices, user can select desired motion peripheral blocks on the MVP graphic tool to customize the design. When new blocks are required, there is unique procedure to allow user to create new block module by reusing existing blocks. Throughout design entry process Verilog coding knowledge is not required with the MVP tool. Figure 8 illustrates the MVP design entry in conjunction with Verilog control block library.

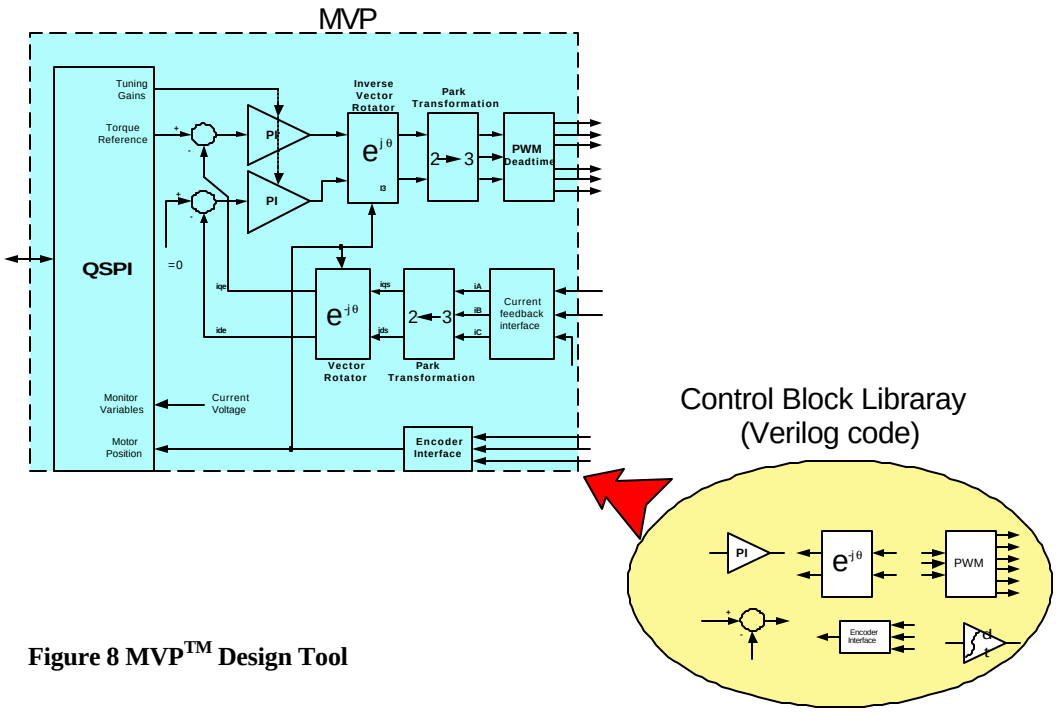

Figure 8 MVP™ Design Tool

System simulation with MVP tool

Unlike traditional approach, simulation process in Accelerator™ system is seamlessly integrated in the whole process of servo drive development. Designer does not have to create parallel simulation path to the real drive system development, and go back and forth between simulation and real drive system testing to cross examine the functionality and performance for design verification. Figure 8 shows Accelerator™ algorithm development flow to illustrates integrated simulation.

Design entry is done only once through Matlab/Simulink™ at the beginning of design. The design data carried out for both simulation and real system all way to the end of design. Designer does not worry about consistency and cohesive synchronization at each phase of design between simulation and real system testing effort.

The design data entered through the MVP tool can be readily downloaded into the FPGA. User can then engage real motor drive testing. At the same time simulation can be performed under Matlab/Simulink™ environment.

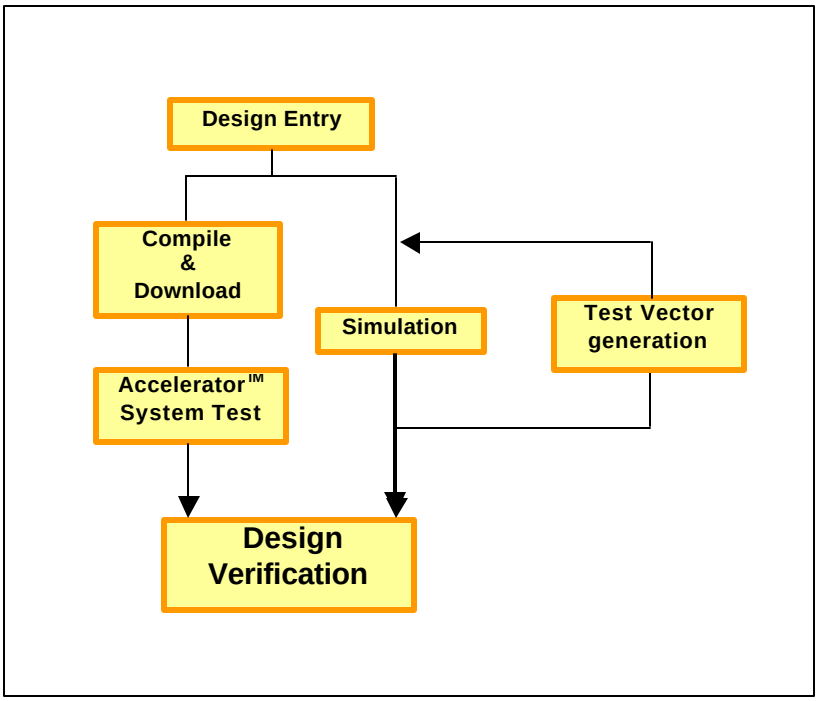

Figure 9 Integrated simulation process of Accelerator™ system

If needed such as military application, user can proceed to ASIC/Gate Array conversion for more optimized solution without significant effort. Since Verilog code is the most popular ASIC language, conversion to ASIC is relatively easy task. The MVP tool will also assist for providing test vectors for ASIC conversion.

Summary

High performance of permanent magnet AC motor servo drive has been presented. The system is implemented in hardware and configurable as opposed to traditional programming approach with DSP or microcontroller. With hardware approach, future performance enhancement such as high bandwidth torque control can be sustained without significant design effort.

New design method based on the MVP tool is also presented to support highly automated design without requiring specific language/coding knowledge. Moreover user can easily customize and uniquely configure specific motion peripherals as a part of configuration of the drive system.

As digital silicon technology and high voltage integrated circuit technology evolve, future high performance servo drive design can be more

easily and quickly realized with the advanced tools such as MVP.

Reference

1. Claus-Ulrich Karipidis, "A Versatile DSP/FPGA Structure optimized for Rapid Prototyping and Digital Real-Time Simulation of Power Electronic and Electrical Drive Systems", Dissertation Thesis, Aachener Beitrage des ISEA, ISBN 3-8265-9738-9.
2. Julio C.G.Pimentel, Hoang Le-Huy, "A VHDL-Based Methodology to Develop High Performance Servo Drivers", IEEE/IAS, 2000 Conference Record.
3. Xilinx Inc.,The Programmable Logic Data Book 2000, Xilinx, 2000.