Specialized Digital Signal Processor for Control of Multi-rail/Multi-phase High Switching Frequency Power Converters

James Mooney, Mark Halton and Abdulhussain E. Mahdi
Department of Electronic & Computer Engineering
University of Limerick
Limerick, Ireland
E-mail: james.mooney@ul.ie

Abstract—This paper describes a Digital Signal Processor (DSP) whose architecture has been purposely developed for executing power converter control algorithms. It is intended for use in the control of multi-phase or multi-rail switching power converters. The DSP’s novel dual multiplier-accumulator datapath allows multiple operations to be executed in a single clock cycle, thus shortening required execution times of power control algorithms compared with standard low-end DSPs. The processor has been designed using the Verilog Hardware Description Language (HDL), implemented on an FPGA development board and tested in a closed-loop system controlling three independent 500 kHz synchronous buck converters.

I. INTRODUCTION

Digital controllers are used in systems where flexibility, programmability and reliability are required. Through the implementation of auto-tuning algorithms, digital controllers can improve power conversion efficiency and maintain accurate voltage regulation by adapting to changes in the power converter’s components over time [1-7]. Three main hardware blocks form the basis for digital power control: an Analog-to-Digital Converter (ADC), a compensator and a Digital Pulse Width Modulator (DPWM). This paper focuses primarily on the digital hardware implementation of the compensator.

Although digital compensators with fixed algorithms have been developed for power converter applications, i.e. Control Law Accelerators (CLAs)) [8-12], in many cases these are unsuitable in practice due to their lack of flexibility. A Digital Signal Processor (DSP)-based compensator is inherently flexible in that it can be programmed to implement a wide range of algorithms, as demonstrated in [13]. Indeed, multiple independent power converters can be simultaneously controlled by a single DSP, executing different control algorithms for each converter [14]. These requirements are common in power supplies for computers and telecommunications systems [15]. However the rate at which the control algorithms must be executed is increasing with the trend towards using higher switching frequencies, higher sampling frequencies and multi-sampling techniques [16-18]. Existing power control DSPs do not have sufficient hardware resources to meet the increasing performance requirements of multi-rail/multi-phase converters with switching frequencies up to and beyond 2 MHz. Using a high-end DSP or multiple CLAs would fulfill these requirements but these are not cost-effective solutions for Point-of-Load (POL) converter applications.

This paper presents an optimal power controller solution with increased computational resources compared with low-end DSPs but without the superfluous resources of a high-end one. The proposed DSP’s architecture has been specifically developed based on analysis of control algorithms and typical multi-rail/multi-phase converter application requirements.

II. DIGITAL SIGNAL PROCESSOR DESIGN

The proposed DSP has three primary components: the program controller, the register file memory and the datapath, as illustrated in Fig. 1. This section describes the unique features of each of these components, which enable the DSP to perform as an effective multi-rail/multi-phase digital power controller. The processor is characterized by its dual datapath architecture, which permits multiple operations to be executed in parallel.

A. Datapath

The datapath component effectively consists of two interconnected datapaths, which receive data from a common register file and share a number of functional elements. Each datapath has a multiplier-accumulator (MAC) unit to execute multiplication, addition/subtraction or combined multiply-accumulate operations in a single DSP clock cycle. These operations are the main operations found in power control algorithms. The execution time of the control algorithms is therefore reduced by carrying out two such operations simultaneously. Data movement operations may also be
executed in parallel with computational operations for updating filter delay lines. Other less frequently required functional elements are shared between the two datapaths. These include barrel shifters that shift data by multiple bits per clock cycle, an Arithmetic Logic Unit (ALU) and saturation logic. The internal saturation logic limits the output of the accumulator registers to the maximum representable value to prevent errors due to overflow. The datapath also includes separate dedicated saturator logic that can limit data values to any given threshold in a single clock cycle operation.

The datapath has a 16-bit fixed-point number representation. This provides sufficient precision for power converter control applications because of the limited range and resolution of the sampled voltage and current signals from the ADC. The resolution of the duty cycle value presented to the DPWM is also typically less than 16 bits [19-21]. The datapath’s accumulator registers have 32-bit resolution in order to preserve accuracy between successive multiply-accumulate operations.

B. Register File Memory

The register file memory stores the coefficients, delay-line values and other data required during the execution of the control algorithms. It is divided into four segments to achieve a concise addressing scheme, which is desirable to minimize the size of the program memory and the instruction decoding hardware. This restricts access to only the selected segment in any single instruction. The segmentation does not have a negative impact on algorithm execution speed in a multiple power converter application because usually only the data associated with one converter needs to be addressed while the algorithm for that converter is executing. A different segment can be selected before executing the algorithm for the next converter by means of a context switch instruction. This register file memory architecture eliminates the need to frequently access data from an external memory source, which would reduce the time available to execute computational operations.

C. Program Controller

The Program Counter (PC) control logic, together with data from a number of configuration registers manage standard instruction sequencing and pipelining hazards such as branching or interrupts. The output of the program counter is the address in the program memory of the next instruction to be executed.

The interrupt control hardware governs how the computational resources of the DSP are time-multiplexed to execute control algorithms for multiple independent power converters. Each control algorithm can be triggered by an external interrupt signal, which may come from the ADC interface, the DPWM or some other timing synchronization hardware block. A control algorithm cannot be interrupted by another interrupt signal and simultaneously occurring interrupts are serviced in terms of their fixed priority setting. This allows converters with different switching frequencies to be controlled by the DSP. The interrupt signal also automatically executes a context switch operation to select the appropriate segment of the register file before any other operation of the algorithm is executed.

When no control algorithm is being executed by the DSP, it operates in background mode, which provides the option of executing monitoring and communications instructions. Control algorithms for power converters with low switching/sampling frequencies can also be executed in this mode. Fig. 2 shows an example of where the DSP executes control algorithms to regulate two independent voltage rails. Background code is executed between each algorithm.

Similar to commercial DSP implementations, an assembly language-based instruction set was created in order to facilitate human programming of control algorithms on the DSP. An example excerpt of code from a control algorithm is illustrated in Fig. 3. Each instruction may contain two independent sub-instructions, which are executed concurrently on separate datapaths. Operations that involve the use of shared datapath elements may only be specified in one of the sub-instructions. An assembler program was also developed to convert the instruction set mnemonics to the binary code that is loaded into the program memory. The program controller’s decoder interprets the binary instructions in the program memory to generate the signals that allow the specified control algorithm to be executed on the processor’s datapath.
Fig. 3. Excerpt from assembly language-based control algorithm with concurrent MAC operations

III. EXPERIMENTAL VERIFICATION

A. Functional Verification

Functional verification of the Verilog-specified DSP design was performed using the ModelSim simulator in conjunction with MATLAB/Simulink/PLECS.

A three-pole, three-zero (3P3Z) control algorithm (1) was used to regulate the output voltage of two independent single-phase buck converter models:

\[
d(n) = b_0 e(n) + b_1 e(n-1) + b_2 e(n-2) + b_3 e(n-3) + a_1 d(n-1) + a_2 d(n-2) + a_3 d(n-3)\]

where \(d(n)\) is the required duty cycle value, \(e(n)\) is the latest sampled output voltage error, \(e(n-1), e(n-2)\) and \(e(n-3)\) are the output voltage errors for one, two and three previous iterations respectively, \(d(n-1), d(n-2)\) and \(d(n-3)\) are the duty cycle values for one, two and three previous iterations respectively and the compensator’s coefficients are \(b_0, b_1, b_2, b_3, a_1, a_2\) and \(a_3\).

Fig. 4 is a screenshot of a selection of the control and data signals involved in executing a single iteration of the 3P3Z algorithm on the DSP. The figure shows that by performing multiple operations in parallel, the processor can execute the algorithm in nine DSP clock cycles. Between successive iterations of the algorithm the DSP is in background mode and can execute monitoring, communications or miscellaneous user defined operations. In this case the processor loops in an idle state in background mode.

B. Digital Hardware Implementation

The Verilog code was synthesized using the integrated synthesizer in Altera’s Quartus II design software, targeting implementation on a Cyclone II FPGA device. The area optimization synthesis technique was selected to minimize logic usage.

Table I compares the utilization of the FPGA’s resources for the DSP’s main constituent elements. In addition to the logic elements and registers listed in the table, the DSP also makes use of some of the FPGA’s embedded hardware components. The DSP’s two 16 x 16 bit multipliers in the datapath are implemented using the FPGA’s embedded multipliers, while the program memory is implemented with the embedded M4K memory blocks.

![DSP control and data signals during simulation of three-pole, three-zero control algorithm](image)
TABLE I. FPGA RESOURCE UTILIZATION FOR SYNTHESIZED DSP CORE

<table>
<thead>
<tr>
<th>Hardware Modules</th>
<th>Logic Elements</th>
<th>% of Total Logic Elements</th>
<th>Registers</th>
<th>% of Total Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register File Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register File</td>
<td>3549</td>
<td>65 %</td>
<td>1296</td>
<td>86 %</td>
</tr>
<tr>
<td>Datapath</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computational Elements</td>
<td>609</td>
<td>11%</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Routing Multiplexers</td>
<td>815</td>
<td>15%</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Program Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decoder</td>
<td>148</td>
<td>3%</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>PC &amp; Interrupt Control</td>
<td>347</td>
<td>6%</td>
<td>213</td>
<td>14%</td>
</tr>
<tr>
<td>Total</td>
<td>5468</td>
<td>100%</td>
<td>1509</td>
<td>100%</td>
</tr>
</tbody>
</table>

The DSP utilizes approximately 16% of the FPGA’s total 33,216 logic elements. The FPGA is intended only as a prototyping platform for the DSP, as it is envisaged that the DSP would ultimately be integrated into a System on Chip (SoC) for multi-rail/multi-phase power converter control. For this reason it necessary to minimize the logic resources of the DSP to minimize the cost of the final SoC solution.

Table I indicates that a major proportion of the logic utilized is allocated to the register file memory block. The register file memory consists of four segments, each with sixteen 16-bit registers. Read access to four registers and write access to two registers is required in each clock cycle. The input de-multiplexers and output multiplexers therefore contribute to a significant proportion of the overall logic requirements. In order to reduce the logic requirements, the number of registers per bank can be reduced. However this confines the DSP to executing basic algorithms, which require fewer coefficients and data variables, resulting in a lower performance compensator solution. An alternative method to reducing the logic requirements is to restrict data access between the two datapath sections. This also leads to reduced flexibility in terms of the possible algorithms that can be executed and can render the DSP unsuitable for power converter control. Investigations into an optimal register file memory architecture are on-going.

The synthesis process yielded a maximum DSP clock frequency of 45 MHz due to the critical path from the register file memory, through the datapath to the accumulator register. Thus a 3P3Z algorithm, which takes nine DSP clock cycles to execute, can be run at a frequency of 5 MHz. This means that a buck converter with a switching frequency of 5 MHz could be controlled by the DSP compensator, where the output voltage is sampled once per cycle. Alternatively, a 1 MHz switching converter could be controlled using over-sampling techniques with an over-sampling factor of five. The DSP could also be used to control three separate converters, each with a switching frequency of 1.66 MHz.

C. Application to DC-DC Converter

The synthesized DSP design was combined with the necessary digital interface hardware to allow data acquisition from the voltage-sampling ADCs. A DPWM was also implemented on the FPGA and interfaced to the processor. The digital system illustrated in Fig. 5 was applied to control a prototype power supply system consisting of three single-phase 12 V - to - 1.5 V buck converters, each with a 500 kHz switching frequency. The DSP was programmed to execute a 3P3Z algorithm to regulate the output voltage of each converter.

Fig. 6 illustrates how the processor is time-multiplexed over the 2 µs switching period to execute the three control algorithms and some background code for testing purposes. In this case the DSP clock frequency has been reduced to 15 MHz to simplify synchronization with the ADC interface. The figure also verifies the ability of the processor to regulate the output voltage of one of the converters in the presence of a load current step. It should be noted that the compensator used here was not designed specifically for the prototype power supply, hence the long settling time that is evident in Fig. 6.
been verified experimentally using an FPGA platform and higher switching frequencies. The operation of the DSP has which will have a larger quantity of rails/phases with demands of future controllers for power supply systems expected that its performance specifications will meet the compared with existing compensator solutions. It is multiple power converters in a cost-effective manner architecture, whose novel design enables it to control DPWM and prototype power supply system.

IV. CONCLUSION

Digital Signal Processors are used in power control applications where controller flexibility is required. This paper has proposed a specialized dual-datapath processor architecture, whose novel design enables it to control multiple power converters in a cost-effective manner compared with existing compensator solutions. It is expected that its performance specifications will meet the demands of future controllers for power supply systems which will have a larger quantity of rails/phases with higher switching frequencies. The operation of the DSP has been verified experimentally using an FPGA platform and a three-rail power converter prototype.

ACKNOWLEDGMENT

Special thanks to Simon Effler, (Circuits and Systems Research Centre, University of Limerick) for his assistance with the experimental verification, by providing the DPWM and prototype power supply system.

REFERENCES