

DSCs Ease Migration to Digital Loop Control

By Bryan Kris, Staff Architect, Architecture & Applications, Digital Signal Controller Division, Microchip Technology, Phoenix

Armed with high execution rates and power peripherals, digital signal controllers provide the functionality needed in power-supply designs but don't require DSP programming.

Application flexibility and cost concerns are creating a demand for intelligent power-supply designs that support soft configuration in production and external control. These advanced power supplies require a digital power-conversion feedback loop. Recent advances in digital signal controllers (DSCs) with high-performance on-chip peripherals targeting power conversion, coupled with their ease of use and affordability, enable many more power-conversion products to migrate to digital loop control.

To implement digital loop control in switch-mode power supply (SMPS) designs using DSCs, designers need to understand the DSC architecture and the factors that influence the selection of the DSC in a particular design. A DSC-based synchronous buck converter will serve as a tutorial example to introduce the design principles associated with developing digital loop control for SMPSs.

Enabling SMPS Designs

When a DSC is used to implement digital loop control in a synchronous buck converter, we arrive at Fig. 1, which describes a typical DSC architecture. DSCs provide the fast-math operations of a DSP, with practical control peripherals and the look and feel of a microcontroller. Control peripherals include counter-based pulse-width-modulation (PWM) modules, analog comparator-based feedback and coordinated analog-to-digital converter (ADC) sampling.

This efficient combination of features and functionality is excellent for power-supply applications, as DSCs provide the high execution rates needed for control-loop software. Even better, complex DSP programming skills are not needed to design using DSCs. Instead, using famil-

iar analog components and microcontroller-like software, designers can use DSCs to quickly and cost-effectively build advanced power supplies. Nevertheless, in this synchronous buck converter SMPS control system example, delays are associated with each function block.

Choosing the correct DSC depends on the particular needs of each application. To choose the right DSC, designers must first select the topology: stepup or stepdown (boost or buck), or isolated (forward, half-bridge or full-bridge). Next, designers must adopt the appropriate switching technique such as hard- or soft-switching and then select a control methodology based on voltage or current mode.

The choice of DSC is determined largely by how well the DSC's peripherals match the desired application. For example, if a soft-switching technique such as zero-voltage transition (ZVT) is desired, then the PWM module onboard the DSC must support dynamic phase shifting. If current-mode control is to be implemented, then analog comparators with reference digital-to-analog converters (DACs) and an ADC with asynchronous-sampling capabilities simplify

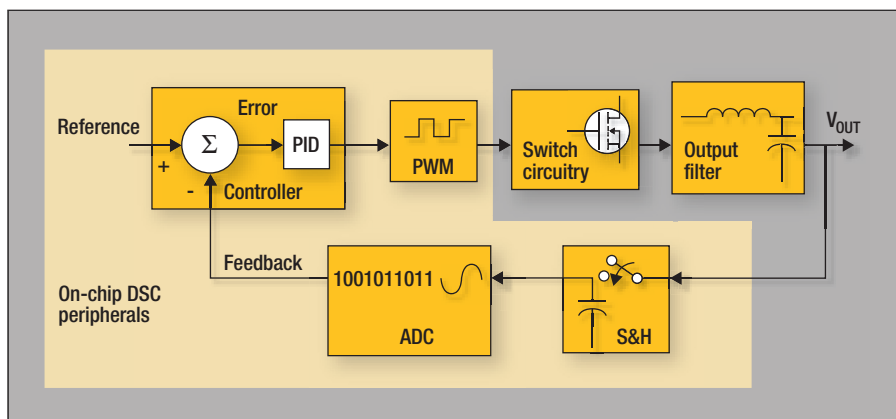


Fig. 1. A typical DSC architecture combines fast-arithmetic operations and control peripherals, as illustrated in this synchronous buck converter.

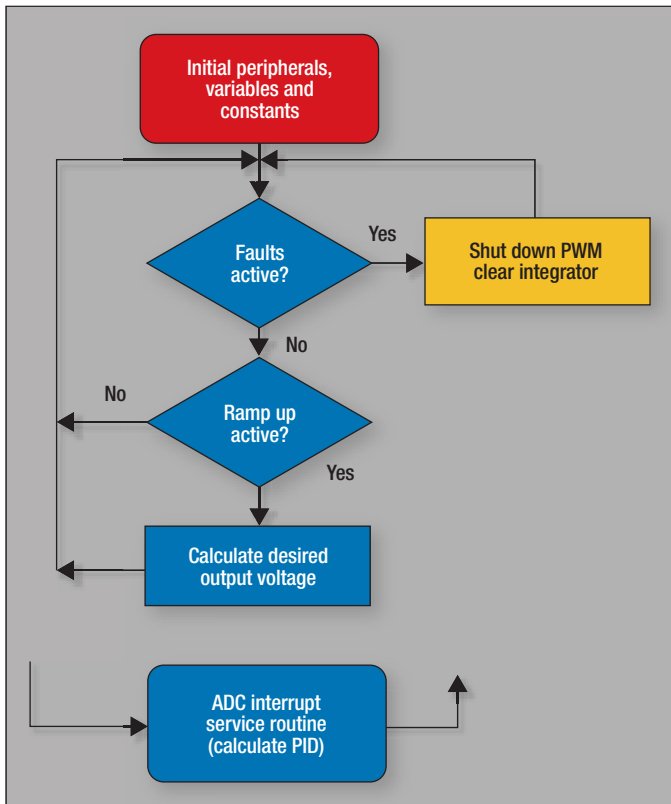


Fig. 2. PID control software structure. The PID loop is interrupt-driven by the ADC on a fixed-time basis.

this task. If the SMPS application is operating at high PWM frequencies to reduce the size and cost of the pc-board components, a high-resolution PWM is needed to reduce voltage and current ripple.

The appropriate PWM operating frequency must be determined, considering that a higher frequency PWM enables the use of smaller inductors and capacitors, but at the cost of additional switching losses.

A key factor to consider when selecting a DSC for this application is to ensure that the onboard PWM module provides adequate resolution for the SMPS design. A resolution that is not high enough will cause the control system to dither the PWM outputs to achieve the desired output, which can create problems with ripple currents and cause the control to enter an unfavorable mode of operation called limit cycling.

Limit cycling is the dithering that the control loop and PWM module cause when the PWM module does not provide sufficient resolution. The control loop will dither the PWM duty cycle, frequency or phase (depending upon the mode of operation) to obtain the desired output voltage or current. This PWM-dithering process occurs at a subcycle basis, where the PWM signal is either greater or less than the needed value. Over a long period of time, the control loop will meet the desired end state, but at a cost that includes increased current and voltage ripple.

How significant is this need for high PWM resolution?

Most DSCs available on the market offer PWM counters that operate in a range from 40 MHz to 150 MHz, yielding PWM resolutions of 6 ns to 25 ns. A PWM with at least 1-ns duty-cycle resolution is needed for digital loop control.

At the operating PWM frequency, the PWM resolution should be comparable to or better than the voltage/current feedback devices' resolution. For example, if the ADC or comparator DAC has 10-bit resolution and the PWM is operating at 1 MHz, then the PWM resolution should also be 10 bits. With a PWM frequency of 1 MHz (1- μ sec period), a PWM generator with 1-ns resolution is capable of subdividing the PWM signal into 1024 pieces, yielding 10-bit resolution.

The ADC onboard a DSC used for digital loop control provides the system with status (feedback) to the control loop. Most ADCs on the market are designed with the assumption that their values are collected and processed in a "group." For example, ADCs used in audio processing and industrial-control systems typically function in this manner. In these cases, group sampling causes the processor workload to peak in groups, which increases control-loop latency.

Often in SMPS circuits, the analog signal to be sampled and converted does not exist, or may not be significant, at all times. The signal may only be important at specific points in the PWM cycle. Therefore, ADC modules that sample in groups may miss the desired data due to imprecise sample timing.

As ADCs cannot continuously monitor signals, samples can only be processed up to the ADC's mega-samples-per-second (MSPS) rating. Some DSCs feature analog comparators that free up the processor and ADC to perform other valuable tasks.

For example, the reference DACs and analog comparators on some DSCs can achieve latencies from current measurement to PWM update of approximately 25 ns. This response time is much faster when compared to other DSCs that rely on "polling" techniques with the ADC and processor to modify the PWM outputs in response to changing conditions. This "nonpolling" method is how the more power-conversion-centric DSCs implement the cycle-by-cycle current-limiting function required for current-mode control.

Implementing current-sensing circuitry in SMPSs is difficult because high currents, fast voltage rise times and pc-board trace inductance often lead to transients in the current-sense circuitry. These transients often occur during the start of the PWM cycle. With the leading-edge blanking (LEB) function, designers can ignore false positives on the fault inputs to the PWM module for a short period of time at the start of a PWM cycle. The inclusion of more fault inputs and modes of operation, combined with LEB support, offers distinct benefits for SMPS designs.

Calculating Control-Loop Delay

Now that the major parameters of the design have been selected, the control-loop timing must be computed to get

an idea of how the synchronous buck converter's power-conversion loop will behave.

In the SMPS design shown in **Fig. 1**, the ADC's sample-and-hold (S&H) circuit typically samples every 2 μ s to 10 μ s. The ADC converts the analog feedback signal to a digital value in approximately 500 ns. A wide variety of control algorithms are available to choose from, including the commonly used proportional, integral and differential (PID) algorithm. On a DSC, the PID controller program runs for about 1 μ s to 2 μ s.

By adding the system's ADC sample-and-convert time (500 ns), PID calculation time (1 μ s), PWM output delay (0), transistor switching time (50 ns) and PID iteration rate period (2 μ s), a total loop delay of 3.65 μ s is derived. This implies a maximum effective control-loop sampling rate of 274 kHz.

While the Nyquist Theorem requires a 2x sampling rate to reconstruct a signal, digital control loops must sample at a 6x to 10x rate. A system with a 2x sampling rate introduces 180 degrees of phase-lag, from the sampling process alone. An 8x sampling rate reduces the phase-lag to 45 degrees of phase.

To maximize phase margin, many digital control systems oversample the analog signals by

10x or more. Assuming a maximum effective sample rate of 274 kHz, the effective control bandwidth is one-eighth of this value, or approximately 34 kHz.

Using the PID algorithm, the proportional, integral and derivative errors of the actual versus the desired output voltage are combined to control the PWM duty cycle. Thus, complex DSP programming skills are not needed to handle the DSP features of controller-centric DSCs (see the code listing in the box).

The central "core" of the control software (**Fig. 2**) is the PID loop, which is interrupt-driven by the ADC on a fixed-time basis. The PID loop is the most time-critical portion of the software and should never exceed approximately 66% of the available processor bandwidth. The remainder of the computing resources can be allocated to the idle-loop software. Typically, functions such as voltage ramp up/down, error detection, feed-forward calculations and communica-

DSP Programming Made Easy

This code listing shows an implementation of the PID algorithm on the dsPIC30F2020 DSC for a digital-mode buck converter. Although the PID software remains the same for most SMPS topologies, the initialization code may need to be modified for the peripheral modules.

```

;These registers are reserved for PID calculations
;w6,w7 = contains data for MAC operations
;w8,w10 = pointers to error terms, and gain coefficients
    COMPUTE_PID:
    push.s                                ; Save SR and W0-W3
    bclr.b IFS0+1,#3                      ; Clear IRQ flag in interrupt controller
    mov #PID_REG_BASE,w8                  ; Initialize pointer to PID register block
    mov #PID_GAIN_REG_BASE,w10           ; Initialize pointer to PID gain register
    block
    mov ADBUF1,w0                          ; Read ADC to get voltage measurement
    mov COMMANDED_VOLTAGE,w1              ; Get commanded output voltage
    sub w1,w0,w0                            ; W0 = proportional voltage error
    mov PROPORTIONAL_ERROR,w1             ; Get previous voltage error
    sub w0,w1,w2                            ; diff error = new verr - old verr
    mov w0,PROPORTIONAL_ERROR             ; Store New Proportional Voltage Error
    mov w0,PREINTEGRAL_TERM               ; Store copy PERR as pre integral term
    mov w2,DERIVATIVE_ERROR               ; Store new Derivative Error
    SUM_PID_TERMS:
    clr A,[w8]+=2,w6,[w10]+=2,w7          ; Clear A, prefetch w6, w7
    mac w6*w7,A,[w8]+=2,w6,[w10]+=2,w7    ; MAC proportional term and gain
    mac w6*w7,A,[w8]+=2,w6,[w10]+=2,w7    ; MAC derivative term and gain
    add ACCA                                ; Update Integrator
    sftac A,-#8                             ; Add ACCB (Integrator) to ACCA
    mov ACCAH,w0                            ; scale accumulator (shift)
    btst ACCAU,#7                           ; Read MSW of acca (result)
    bra z,OUTPUT_PWM                       ; Check sign bit of ACCA
    clr w0                                    ; Branch if acca PWM value is positive
    OUTPUT_PWM:
    mov w0,DC1                              ; Clear negative PWM values
    pop.s                                    ; Output new duty-cycle value
    retfie                                  ; Restore SR, w0-w3
    ; Return from Interrupt

```

tion-support routines must be performed in the idle loop.

Assuming 30 MIPS operation with the PID loop (comprising 30 instructions, the execution time is approximately 1 μ s), if the iteration rate is 500 kHz (2 μ s), then the PID workload consumes one-half of the available processor bandwidth, or 15 MIPS.

The power-stage components used in SMPS designs greatly affect the overall performance and reliability of the designs. In the case of the synchronous buck converter example, the selection of components such as the MOSFETs, inductor, and input and output capacitors will entail the same calculations as when a conventional analog control loop is employed.

For those interested in building the power-supply design presented here, the online version of this article provides a compressed file containing source listings, a hex file, Gerber files and a bill of materials.

PETech