Embedded University

Digital Power Regulation Solutions

Instructor:

Tom Spohrer,

Product Marketing Manager,

MCU16 Division, Microchip Technology, Inc.

MICROCHIP

Powered by Embedded COMPUTING DESIGN

Embedded University

HOUSEKEEPING

- Housekeeping
- Presentation
- Text Chat Questions and Answers
- Wrap-up

MICROCHIP

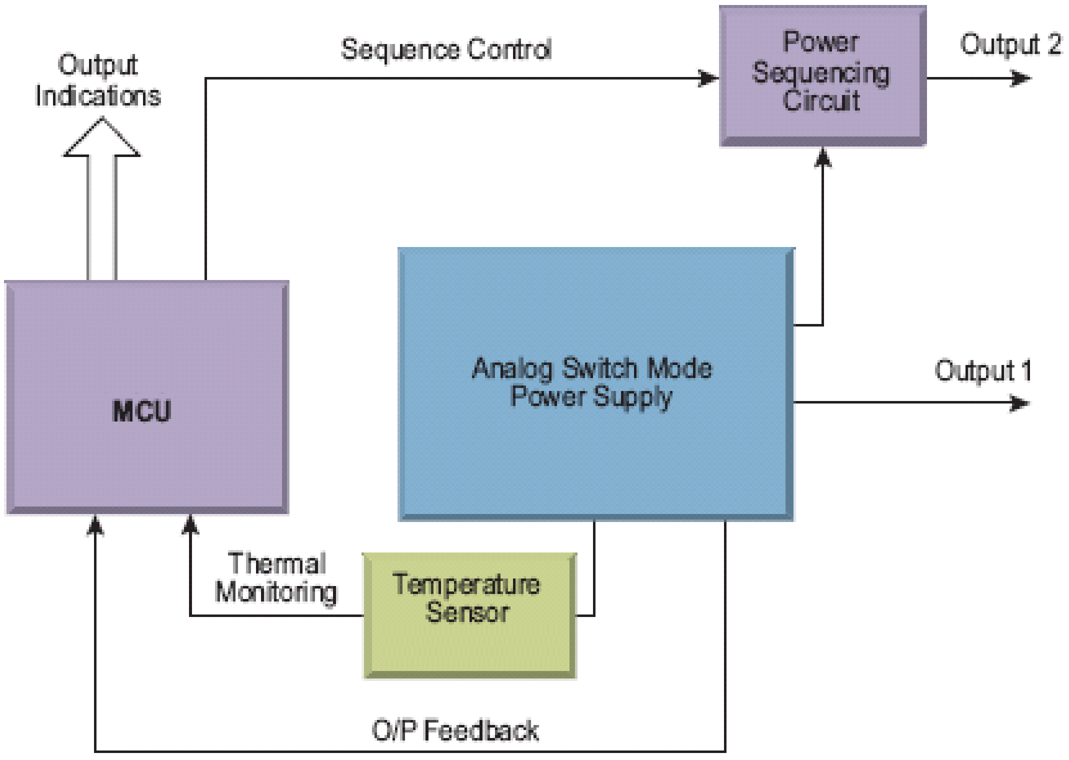
Powered by Embedded COMPUTING DESIGN

Agenda

- **Review the Levels of Digital Integration**
- **Benefits of Full Digital Control**
- **Digital Controller Basics**
  - Analog SMPS and Digital SMPS Implementations
- **The Digital PID**
  - The Mathematics, Generating the Coefficients, DSC Digital PID Implementation
- **Typical DSC Firmware Architecture**
- **Other Digital Compensator Types**
- **Advanced Digital Control**
  - Adaptive, Non-linear, and Predictive algorithms
- **Additional Resources**
  - App Notes, Libraries, Digital Compensator Design Tool, Workshops

Levels of Digital Integration

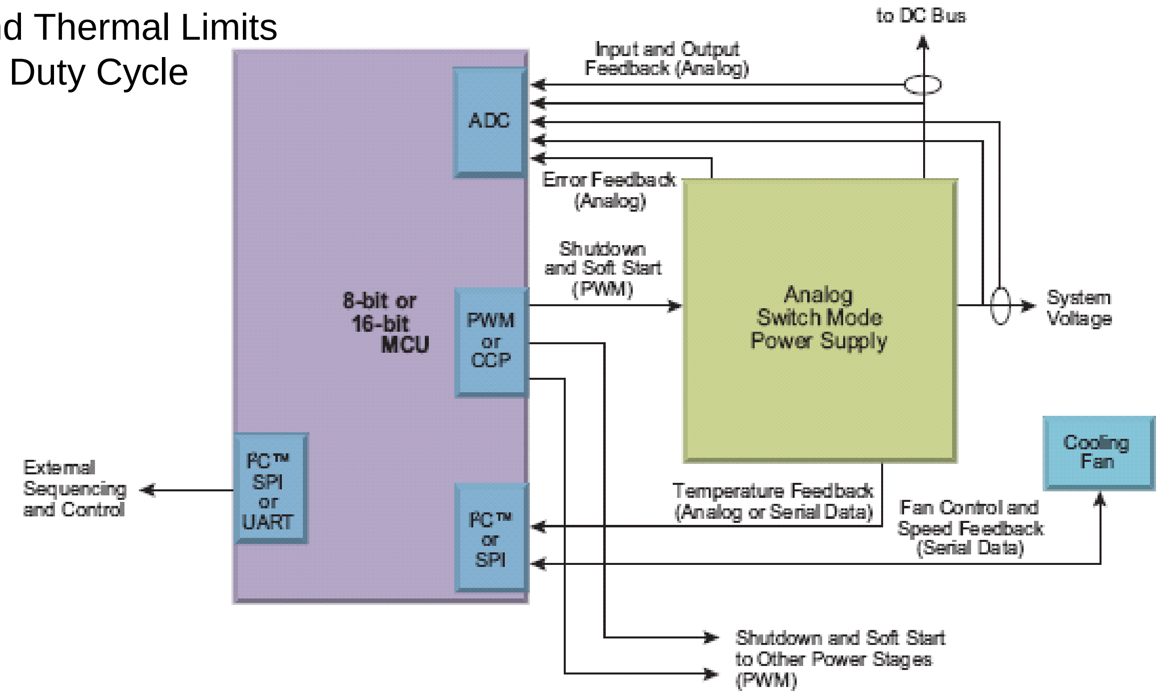
- **<u>Level 1</u>: Control improves traditional analog power design**
  - Soft-start and Power Sequencing
  - Voltage and Temperature Monitoring
  - Communication and Data Logging



Levels of Digital Integration

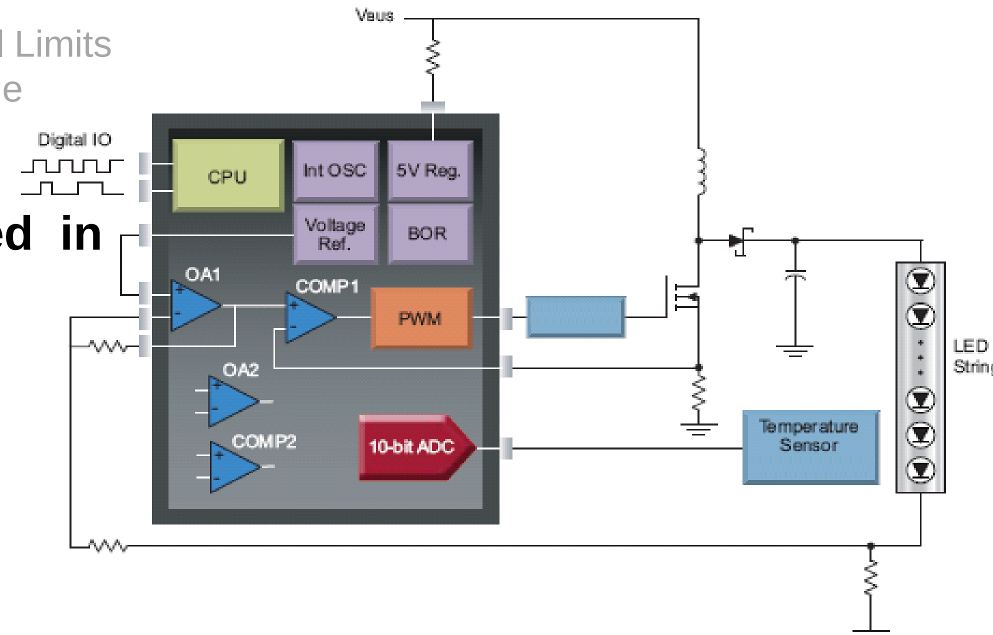
- Level 1: Control improves traditional analog power design
  - Soft-start and Power Sequencing
  - Voltage and Temperature Monitoring
  - Communication and Data Logging

- Level 2: MCU controls reference signals for the PWM Controller
  - Indirectly Set Voltage, Current and Thermal Limits
  - Determine PWM Period and Max Duty Cycle
  - Self-calibration



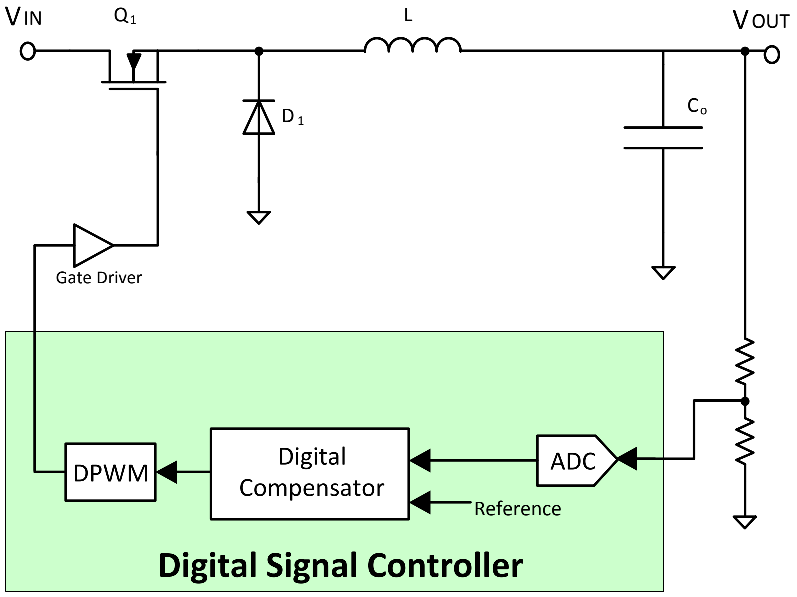
Levels of Digital Integration

- <u>Level 1</u>: Control improves traditional analog power design
  - Soft-start and Power Sequencing
  - Voltage and Temperature Monitoring
  - Communication and Data Logging
- <u>Level 2</u>: MCU controls reference signals for the PWM Controller
  - Indirectly Set Voltage, Current and Thermal Limits
  - Determine PWM Period and Max Duty Cycle
  - Self-calibration
- **<u>Level 3</u>: Analog functions integrated in MCU and can be reconfigured for changing load conditions**
  - Micro Controls the Feedback and PWM



Levels of Digital Integration

- **Level 1: Control improves traditional analog power design**
  - Soft-start and Power Sequencing
  - Voltage and Temperature Monitoring
  - Communication and Data Logging
- **Level 2: MCU controls reference signals for the PWM Controller**
  - Indirectly Set Voltage, Current and Thermal Limits
  - Determine PWM Period and Max Duty Cycle
  - Self-calibration
- **Level 3: Analog functions integrated in MCU and can be reconfigured for changing load conditions**
  - Micro Controls the Feedback and PWM
- **Level 4: Analog feedback is replaced with complete digital DSP-based control**
  - PID, 2P2Z, 3P3Z Algorithms with Voltage or Current Mode Control



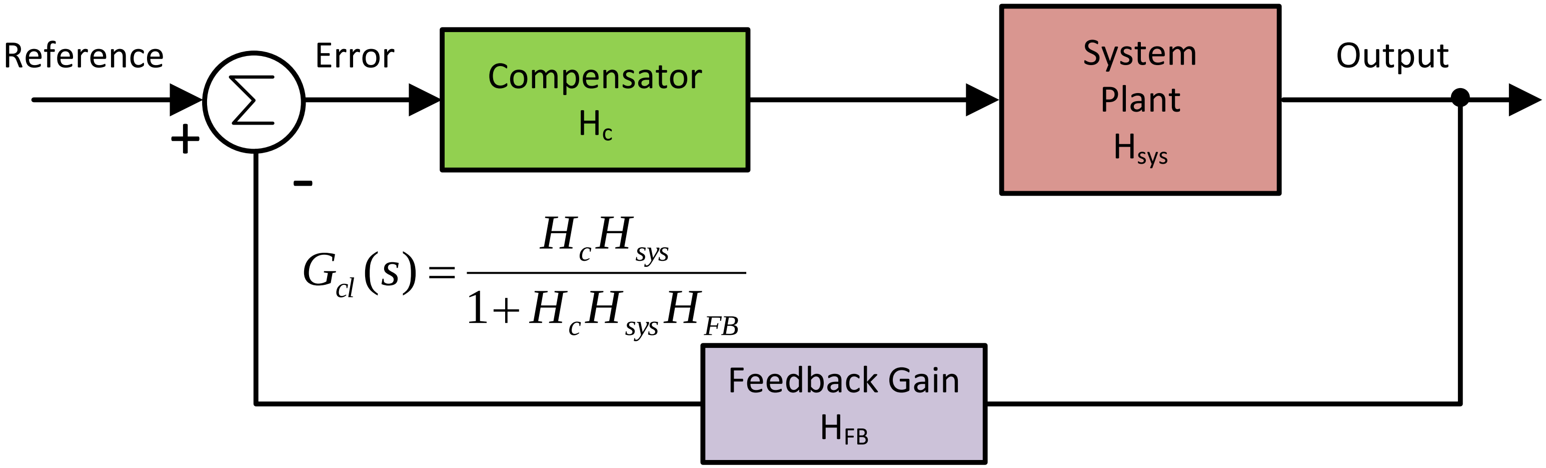
Technical Advantages of Digital Control

- Software-driven controllers can implement any topology
- Non-linear, predictive, and adaptive control techniques can be implemented for highest efficiency across widely varying load and environmental conditions
- Change topology "On-the-Fly": e.g. multi-phase to single phase at low-load to optimize efficiency
- Minimize over-specification of magnetic components because DSC provides tighter tolerance than passive Rs & Cs
- Reduced BOM costs
- Fewer components result in higher power densities and higher reliability

Business Advantages of Digital Control

- IP belongs to PS vendor & resides in secured firmware
- Fewer hardware platforms support more products
- Easily update software to meet changing customer needs
- Improved self-test capability simplifies product testing
- Component tolerance & drift reduced:
  (Eliminate “Over specified” components)
- Restrict products from operating beyond specification
- Log any misuse in the advent of warranty return

Controller Basics



$$G_{cl}(s) = \frac{H_c H_{sys}}{1 + H_c H_{sys} H_{FB}}$$

Reference: This is the desired set-point we want the Output to follow

Error: Calculation of (Reference – Feedback), this is the value the compensator acts upon

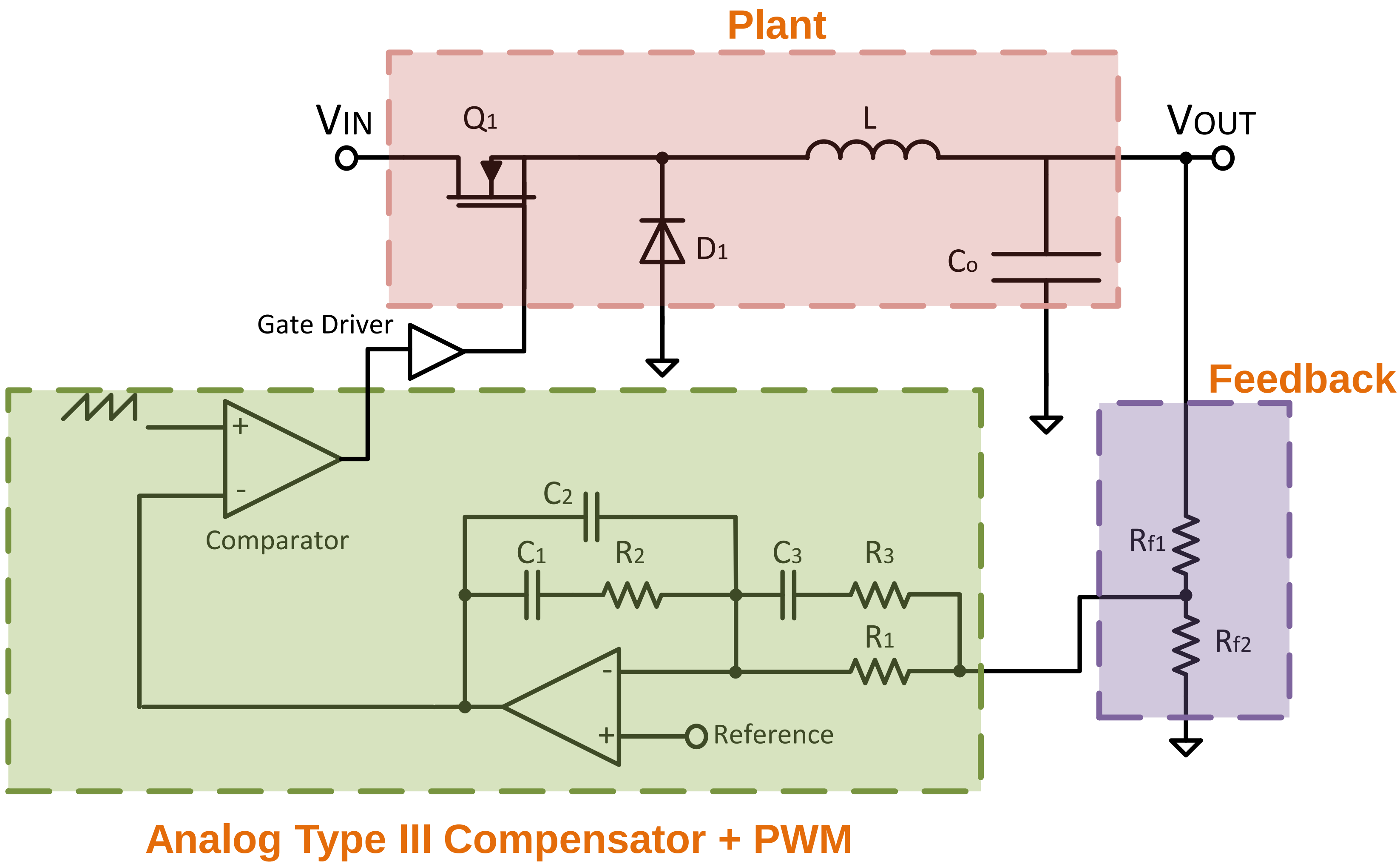
Compensator: Digital compensator (PID, 2P2Z, 3P3Z, or 4th order, etc.)

System: This is the System/Plant/Power stage being controlled (SMPS, Motor, Actuator, etc)

Output: For our purposes this will be Voltage or Current

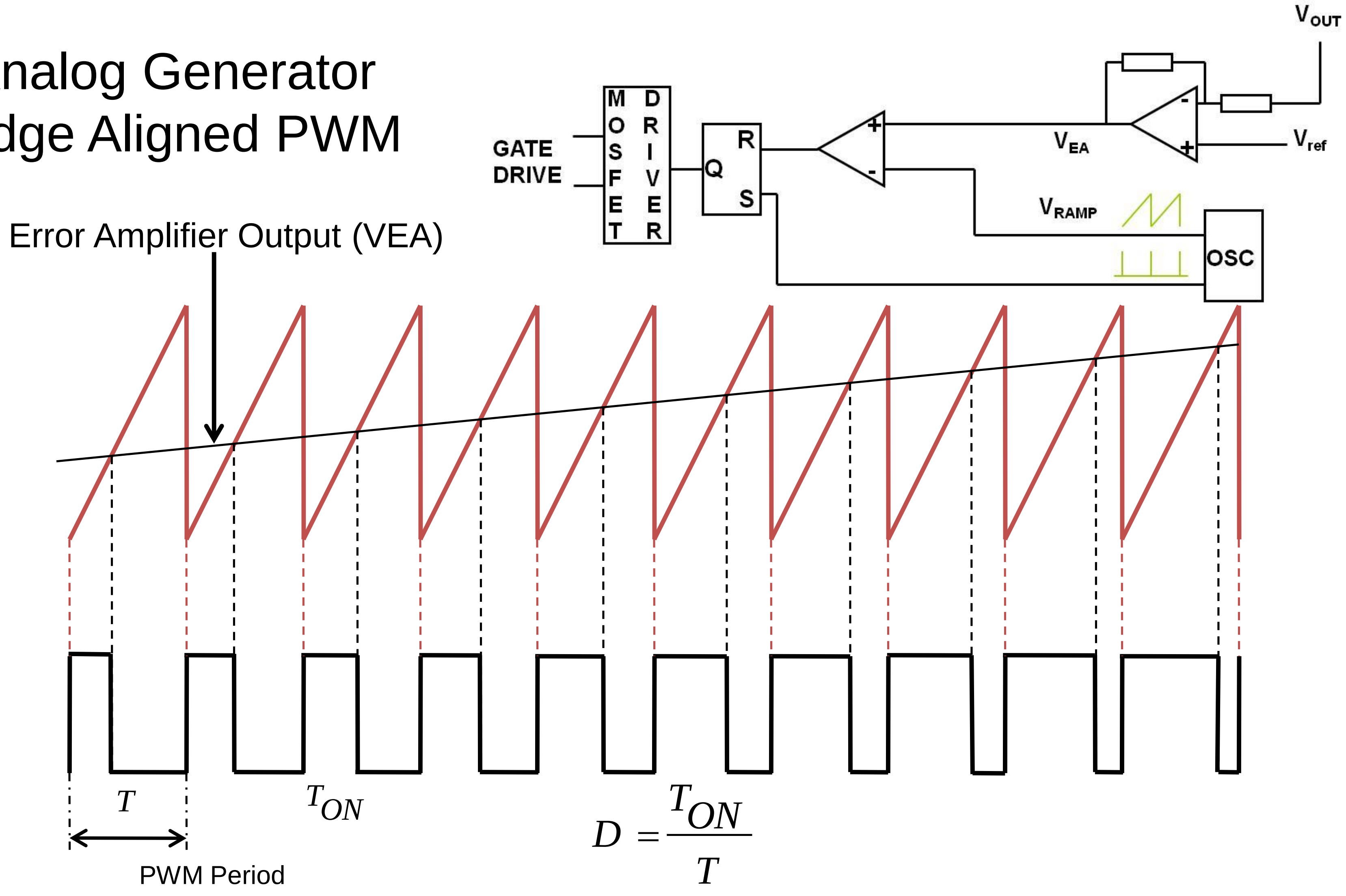
Feedback: The measurement of the output signal level

Analog PWM SMPS Implementation



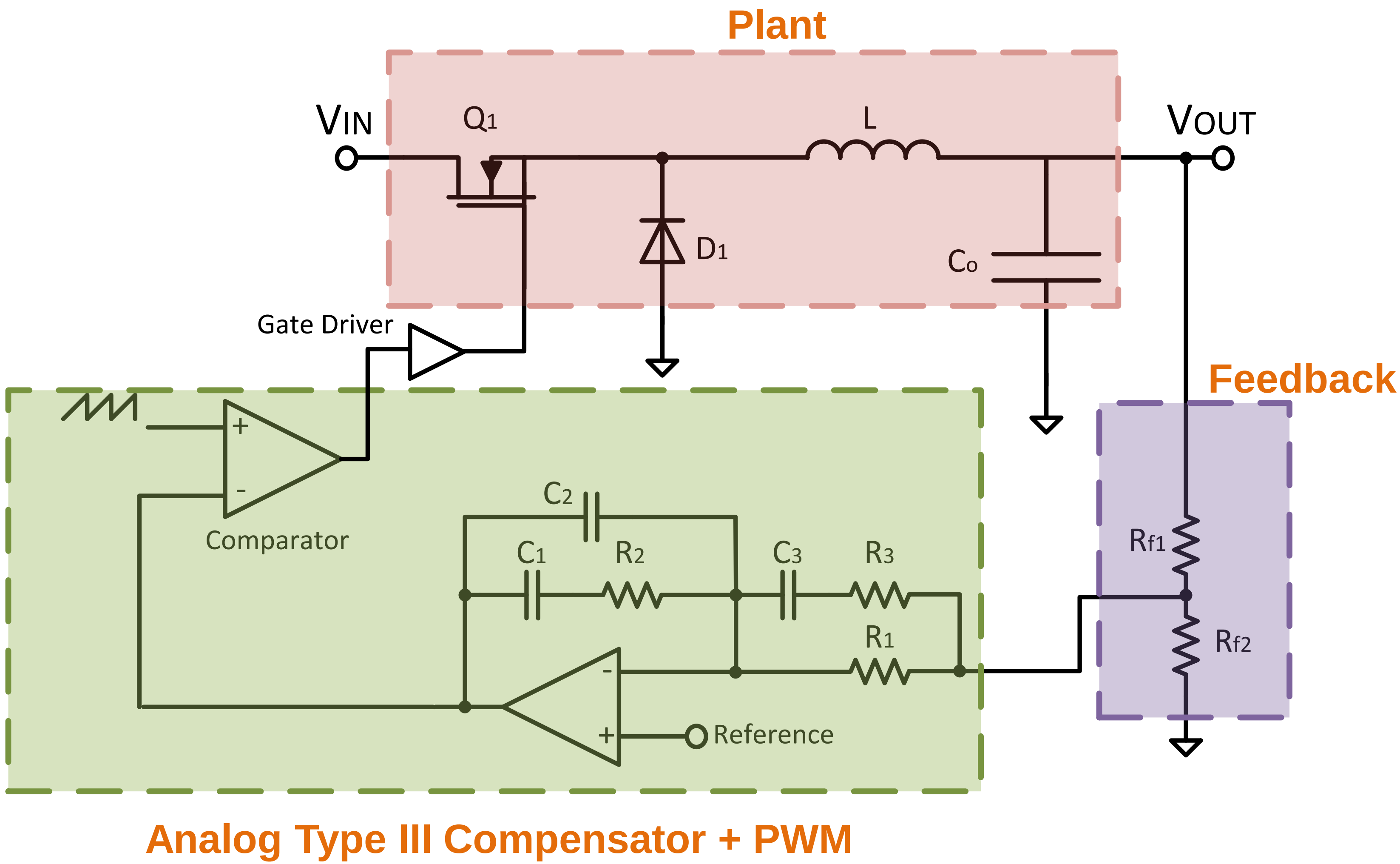
Analog PWM Generator

Analog Generator
Edge Aligned PWM

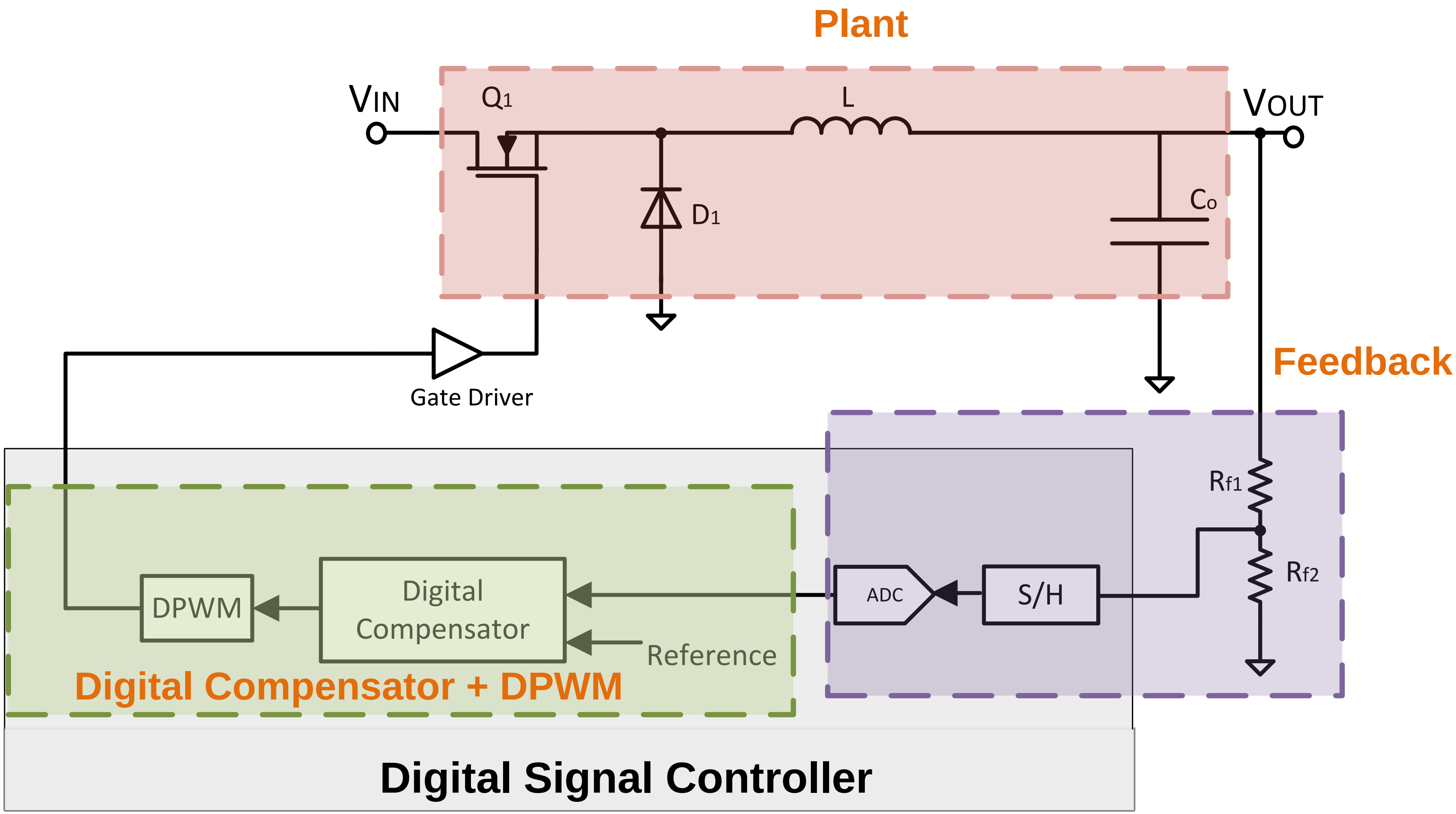


$$D = \frac{T_{ON}}{T}$$

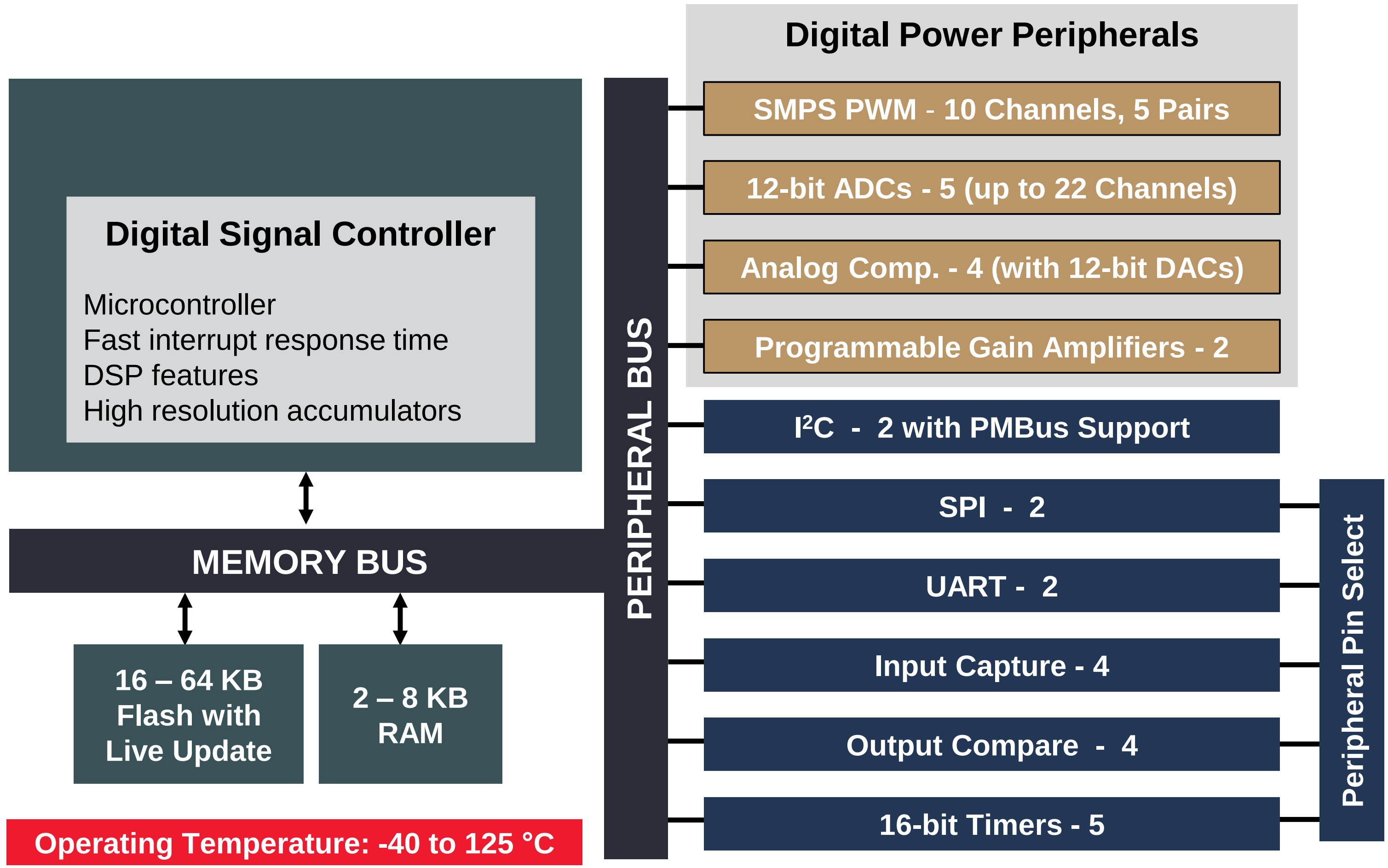
Analog PWM SMPS Implementation



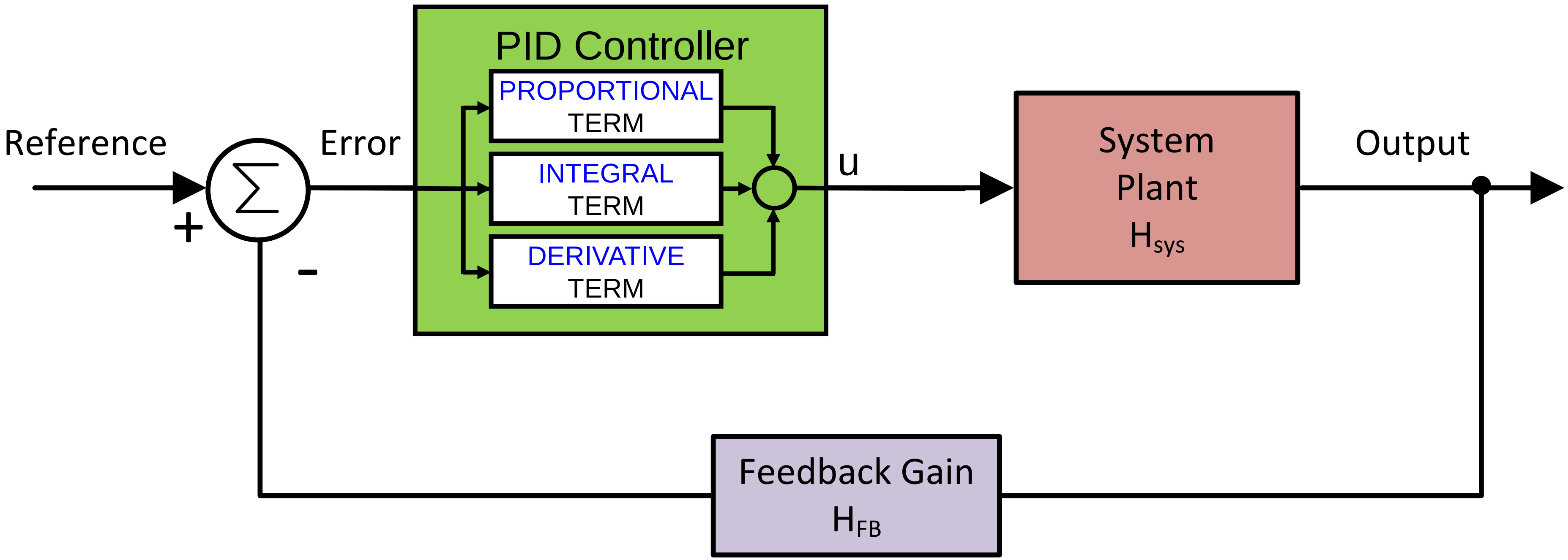
Digital PWM SMPS Implementation



Example Digital Signal Controller for Digital Power



The PID Controller



PID Controller Mathematics

PID Implementation

- From: $R_{PID}^{BE}(s) = k_p + \frac{k_i}{s} + k_d\, s$

PID Controller Mathematics

PID Implementation

- From: $$R_{PID}^{BE}(s) = k_p + \frac{k_i}{s} + k_d\, s$$

- Using: $$s = \frac{z-1}{T\,z}$$ (Backward Euler)

PID Controller Mathematics

PID Implementation

- From: $$R_{PID}^{BE}(s) = k_p + \frac{k_i}{s} + k_d\, s$$

- Using: $$s = \frac{z-1}{T\,z}$$ (Backward Euler)

- We get: $$R_{PID}^{BE}(z) = \frac{\left(k_p + k_i\,T + \frac{k_d}{T}\right) + \left(-k_p - 2\,\frac{k_d}{T}\right) z^{-1} + \left(\frac{k_d}{T}\right) z^{-2}}{\left(1 - z^{-1}\right)}$$

PID Controller Mathematics

PID Implementation

Z-Domain:

$$R_{PID}^{BE}(z) = \frac{\left(k_p + k_i T + \frac{k_d}{T}\right) + \left(-k_p - 2\frac{k_d}{T}\right) z^{-1} + \left(\frac{k_d}{T}\right) z^{-2}}{\left(1 - z^{-1}\right)}$$

Time Domain:

$$u(n) = u(n-1) + \left(k_p + k_i T + \frac{k_d}{T}\right) e(n) + \left(-k_p - 2\frac{k_d}{T}\right) e(n-1) + \left(\frac{k_d}{T}\right) e(n-2)$$

PID Controller Mathematics

PID Implementation

Z-Domain:

$$R_{PID}^{BE}(z)=\frac{\left(k_p+k_i T+\frac{k_d}{T}\right)+\left(-k_p-2\frac{k_d}{T}\right)z^{-1}+\left(\frac{k_d}{T}\right)z^{-2}}{\left(1-z^{-1}\right)}$$

Time Domain:

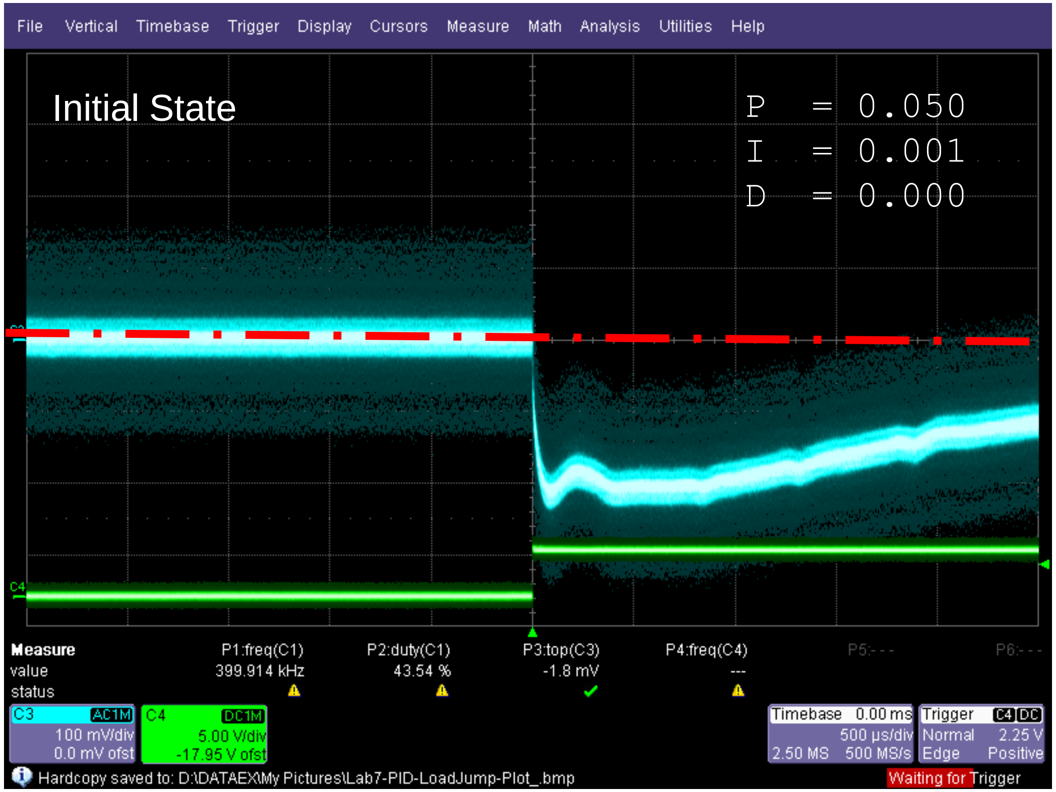
$$u(n)=u(n-1)+\underbrace{\left(k_p+k_i T+\frac{k_d}{T}\right)}_{K_A}e(n)+\underbrace{\left(-k_p-2\frac{k_d}{T}\right)}_{K_B}e(n-1)+\underbrace{\left(\frac{k_d}{T}\right)}_{K_C}e(n-2)$$

Constants

Finding PID Coefficients Ziegler-Nichols Method

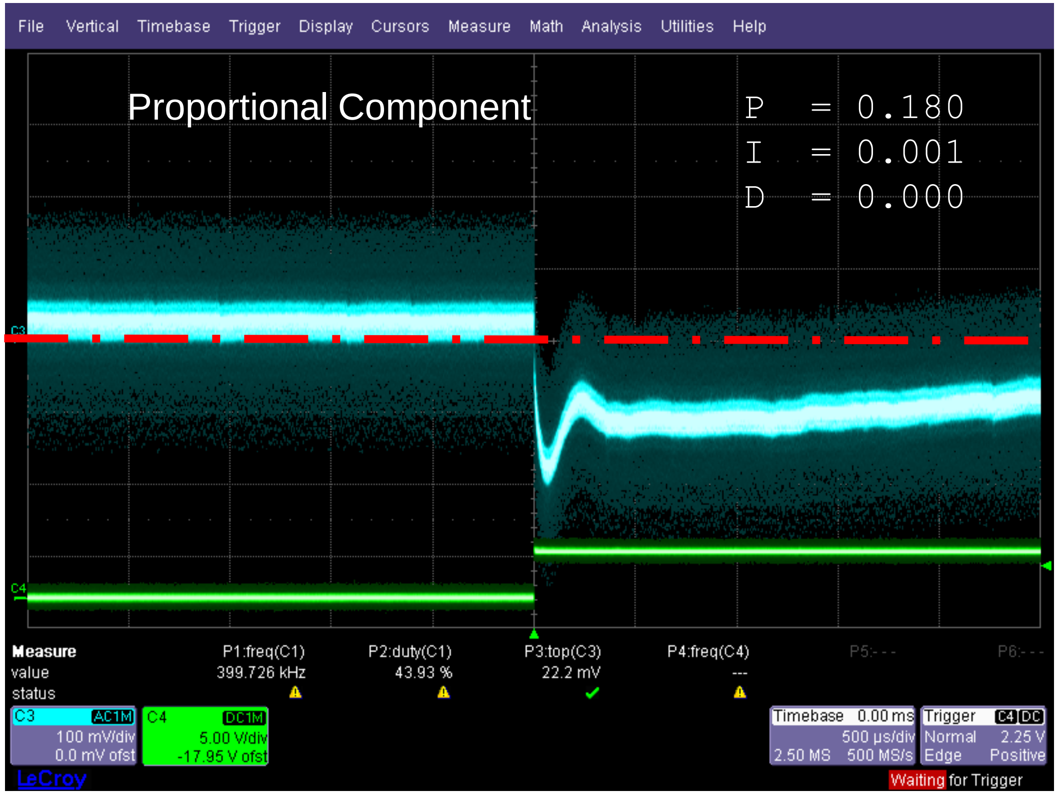
1) Start with K_P and K_I coefficients near to (but not) zero. (K_D can be 0)

2) Allow system to stabilize, then introduce a step change in the load



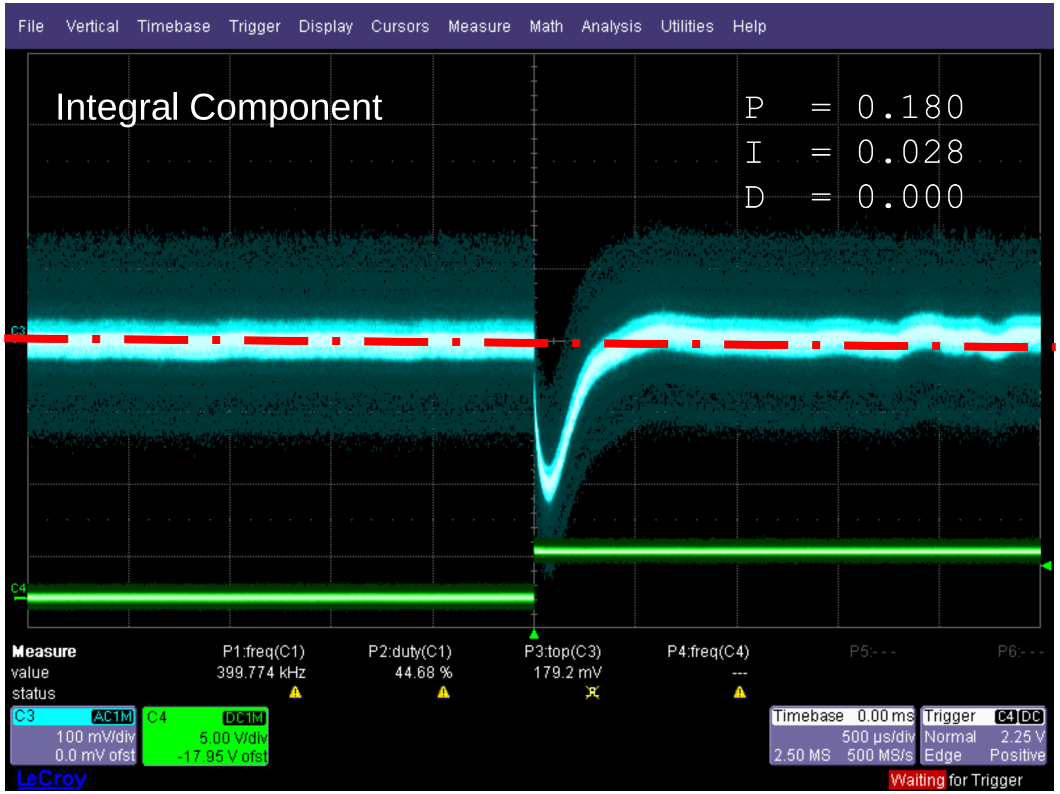
Finding PID Coefficients
Ziegler-Nichols Method

3) Increase K_P until the tiny overshoot reaches the output voltage level and then reduce it to approx. 70% of its value



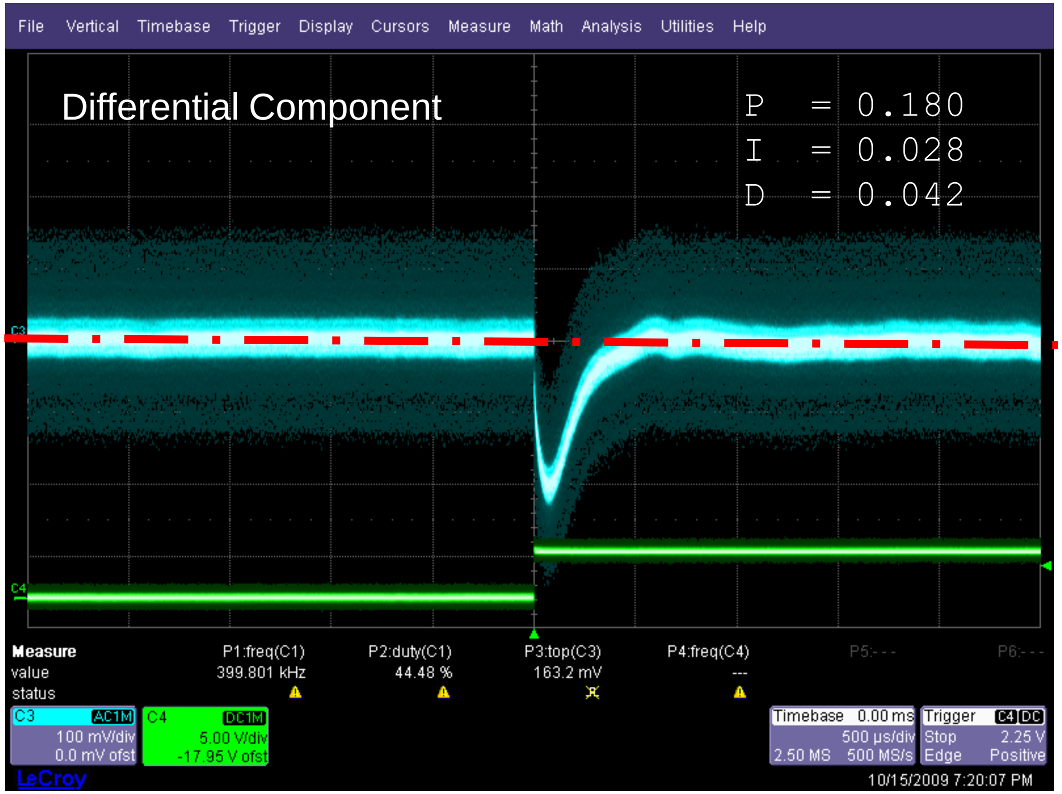
Finding PID Coefficients
Ziegler-Nichols Method

4) Increase K_I to get a nice response



Finding PID Coefficients
Ziegler-Nichols Method

5) Increase K_D reduce oscillations during the load-condition
(Keep K_D low to avoid noise susceptibility)



PID Controller Mathematics

PID Implementation

Z-Domain:

$$R_{PID}^{BE}(z)=\frac{\left(k_p+k_i T+\frac{k_d}{T}\right)+\left(-k_p-2\frac{k_d}{T}\right)z^{-1}+\left(\frac{k_d}{T}\right)z^{-2}}{\left(1-z^{-1}\right)}$$

Time Domain:

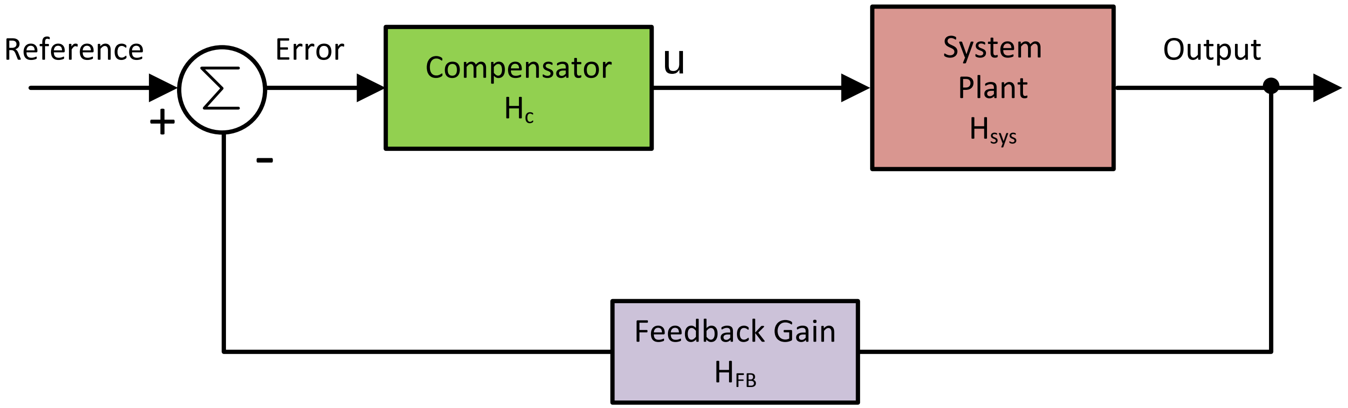
$$\underbrace{\quad}_{}\qquad K_A \qquad K_B \qquad K_C$$

$$u(n)=u(n-1)+\left(k_p+k_i T+\frac{k_d}{T}\right)e(n)+\left(-k_p-2\frac{k_d}{T}\right)e(n-1)+\left(\frac{k_d}{T}\right)e(n-2)$$

Constants

Finding the Coefficients Using a Model

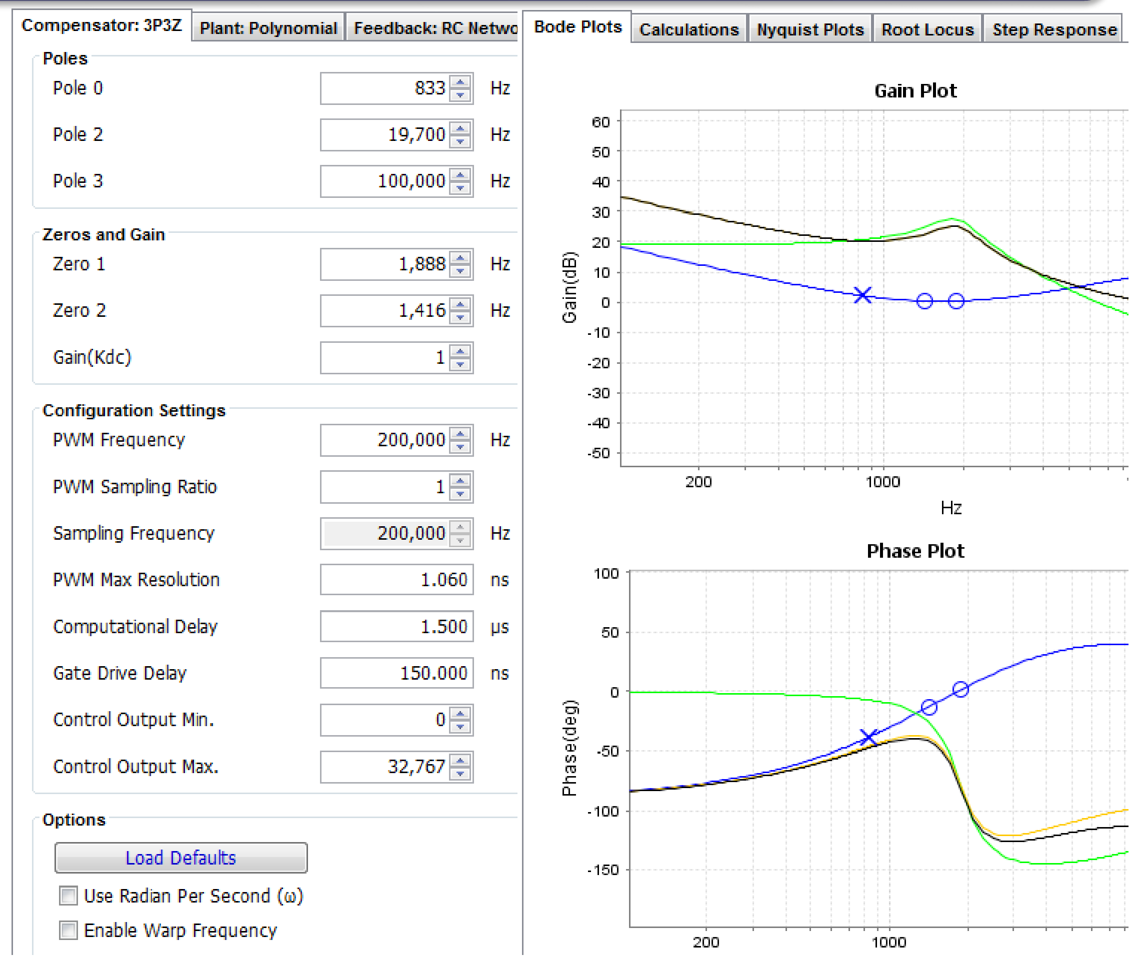
- Circuitry can be modeled and coefficients calculated using simulation tools like MATLAB
- Characterize power circuitry to get equation for the system plant output in terms of the controlled variable (u)
- The compensator equation is known – an output of the controlled variable (u) in terms of the process output
- Substitute the process equation into the compensator equation and solve for the coefficients using simultaneous equations



Finding the Coefficients Using a Tool - DCDT

Eliminate the need for manual bi-linear transform calculations

- **Input Plant Details**
  - Polynomial or Pole/Zero form
  - Imported data table from simulation or network analyzer
- **Five Compensator Types**
  - Digital 2P2Z, Digital 3P3Z
  - PID, Analog Type II, III
- **Generate header file with compensator coefficients**
- **Analyze response & stability**



PID Controller Mathematics

PID Implementation

Z-Domain:

$$R_{PID}^{BE}(z) = \frac{\left(k_p + k_i T + \frac{k_d}{T}\right) + \left(-k_p - 2\frac{k_d}{T}\right) z^{-1} + \left(\frac{k_d}{T}\right) z^{-2}}{\left(1 - z^{-1}\right)}$$

K_A K_B K_C

Time Domain:

$$u(n) = u(n-1) + \left(k_p + k_i T + \frac{k_d}{T}\right) e(n) + \left(-k_p - 2\frac{k_d}{T}\right) e(n-1) + \left(\frac{k_d}{T}\right) e(n-2)$$

Constants

MAC Instruction

One instruction,
One clock cycle,
8 operations

MAC w6*w7, A, [w8]+=2, w6, [w10]-=4, w7, [w13]+=2

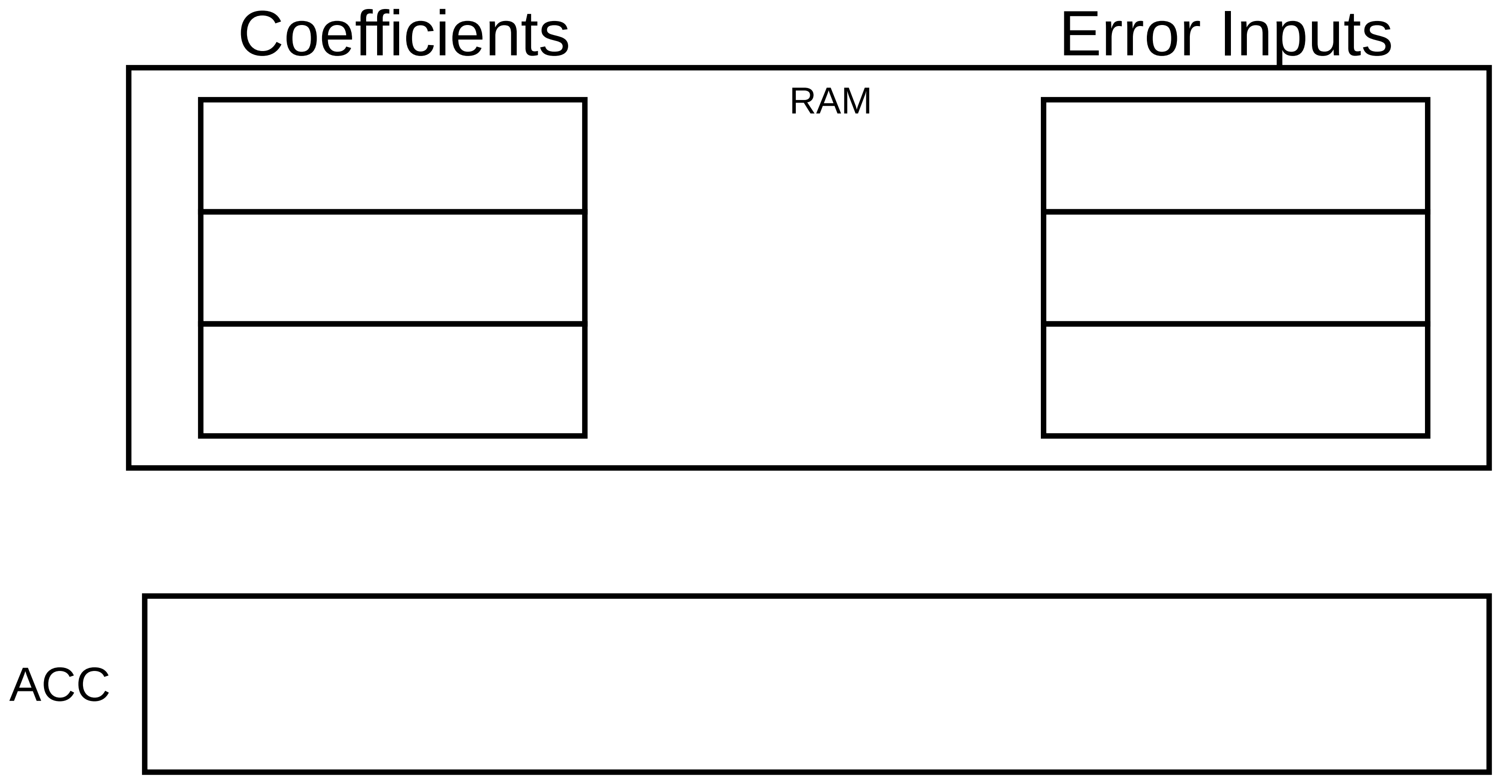
One instruction performs:

```
A = W6 * W7      ; W6 multiplied by W7 and product added to A
W6 = (W8)        ; load new data addressed by W8 into W6
W7 = (W10)       ; load new data addressed by W10 into W7
W8 = W8+2        ; Add 2 to address in W8
W10 = W10-4      ; Subtract 4 from address in W10
(W13) = B        ; Copy B (rounded) to memory specified by W13
W13 = W13+2      ; Increment W13 by 2
```

One instruction, One clock cycle

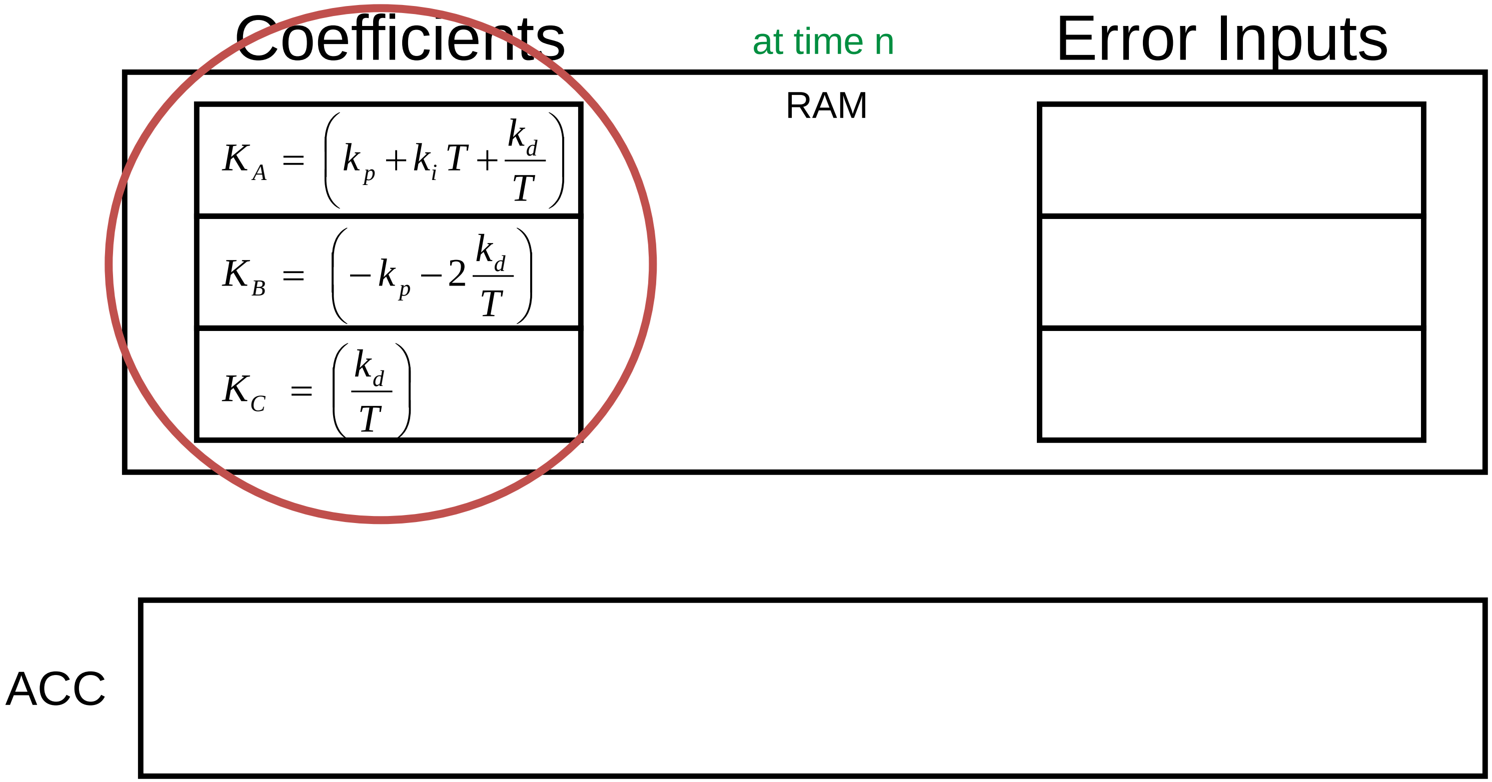
DSC Digital PID

PIDs in DSCs: Architecture



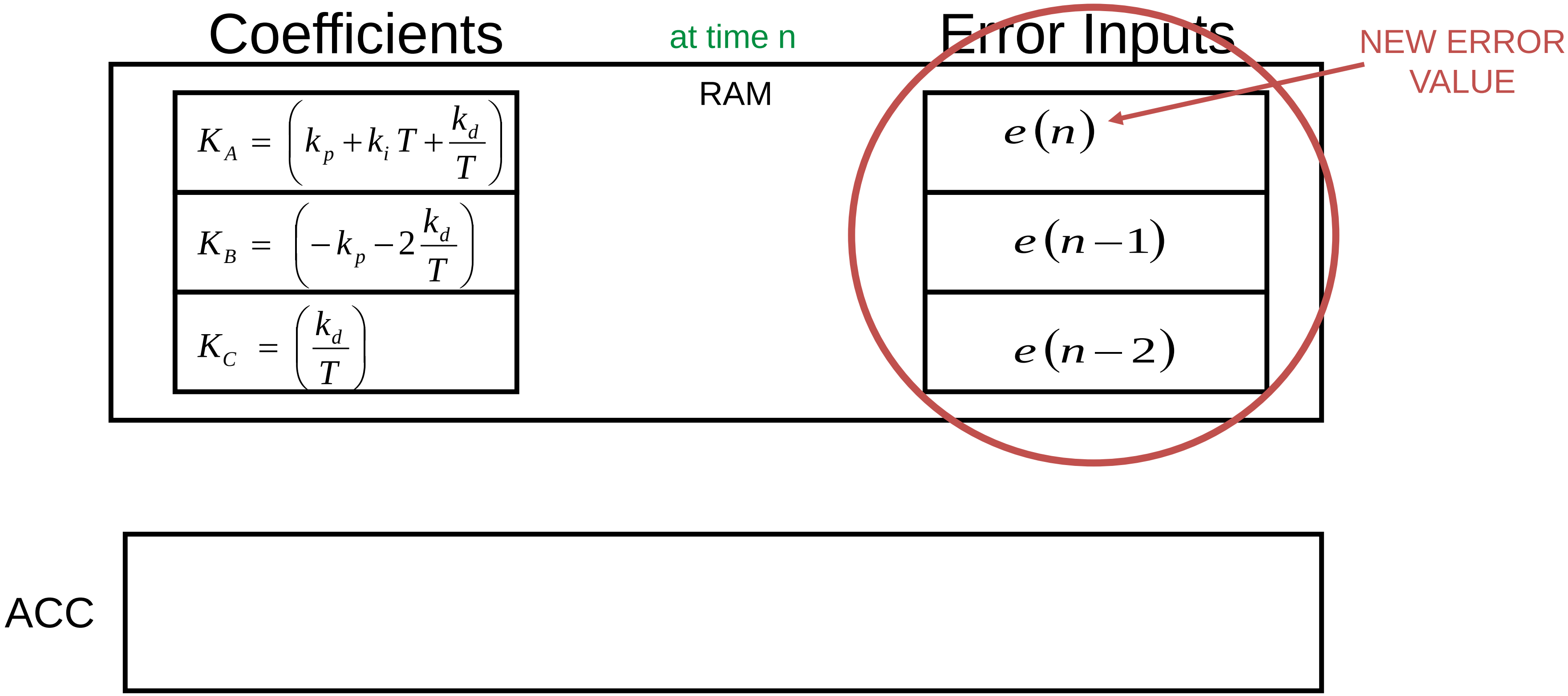
DSC Digital PID

PIDs in DSCs: Coefficients



DSC Digital PID

PIDs in DSCs: New Error Value



DSC Digital PID

PIDs in DSCs: Previous result

Coefficients | at time n | Error Inputs

RAM

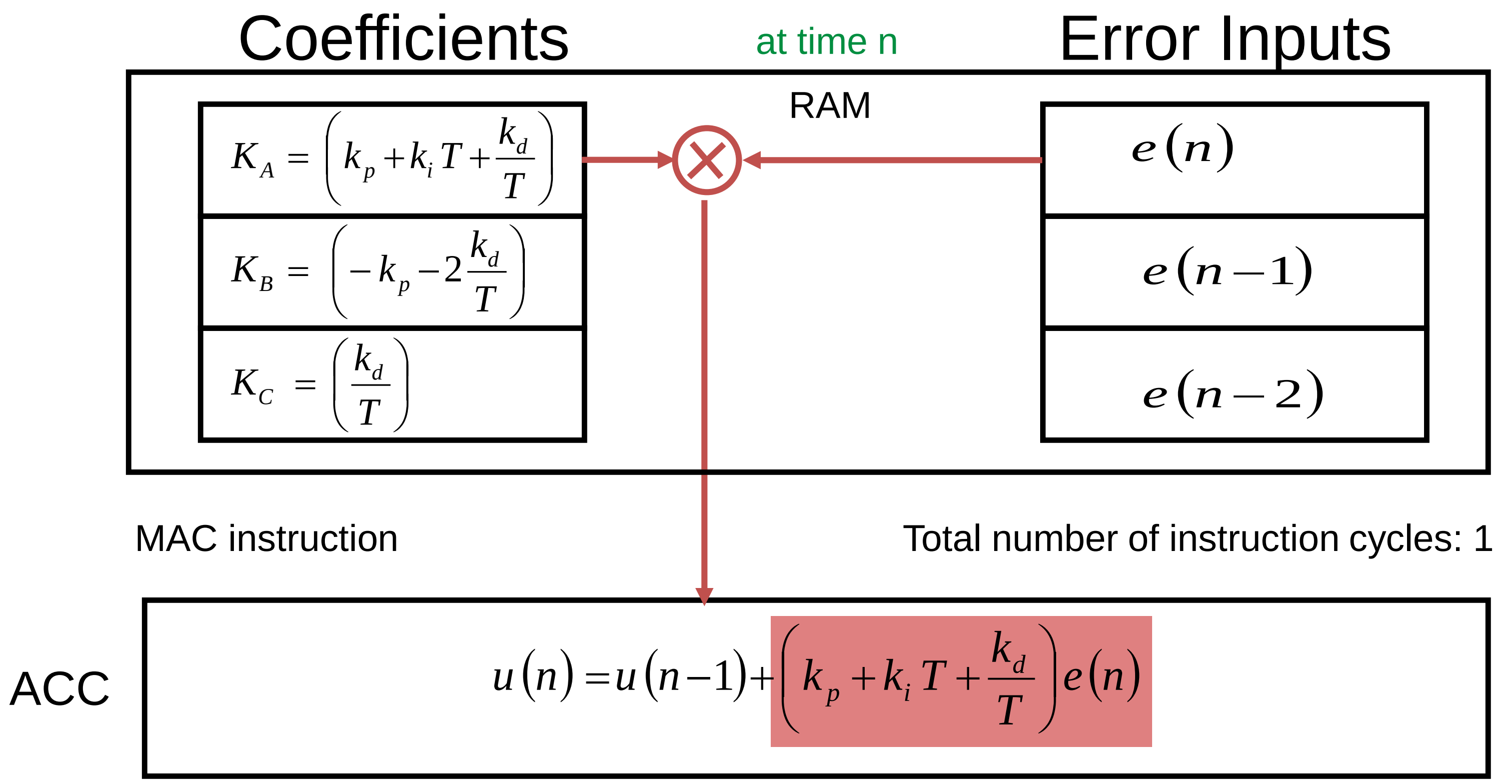
Coefficients	Error Inputs
$K_A = \left(k_p + k_i T + \frac{k_d}{T}\right)$	$e(n)$
$K_B = \left(-k_p - 2\frac{k_d}{T}\right)$	$e(n-1)$
$K_C = \left(\frac{k_d}{T}\right)$	$e(n-2)$

ACC

$$u(n) = u(n-1)$$

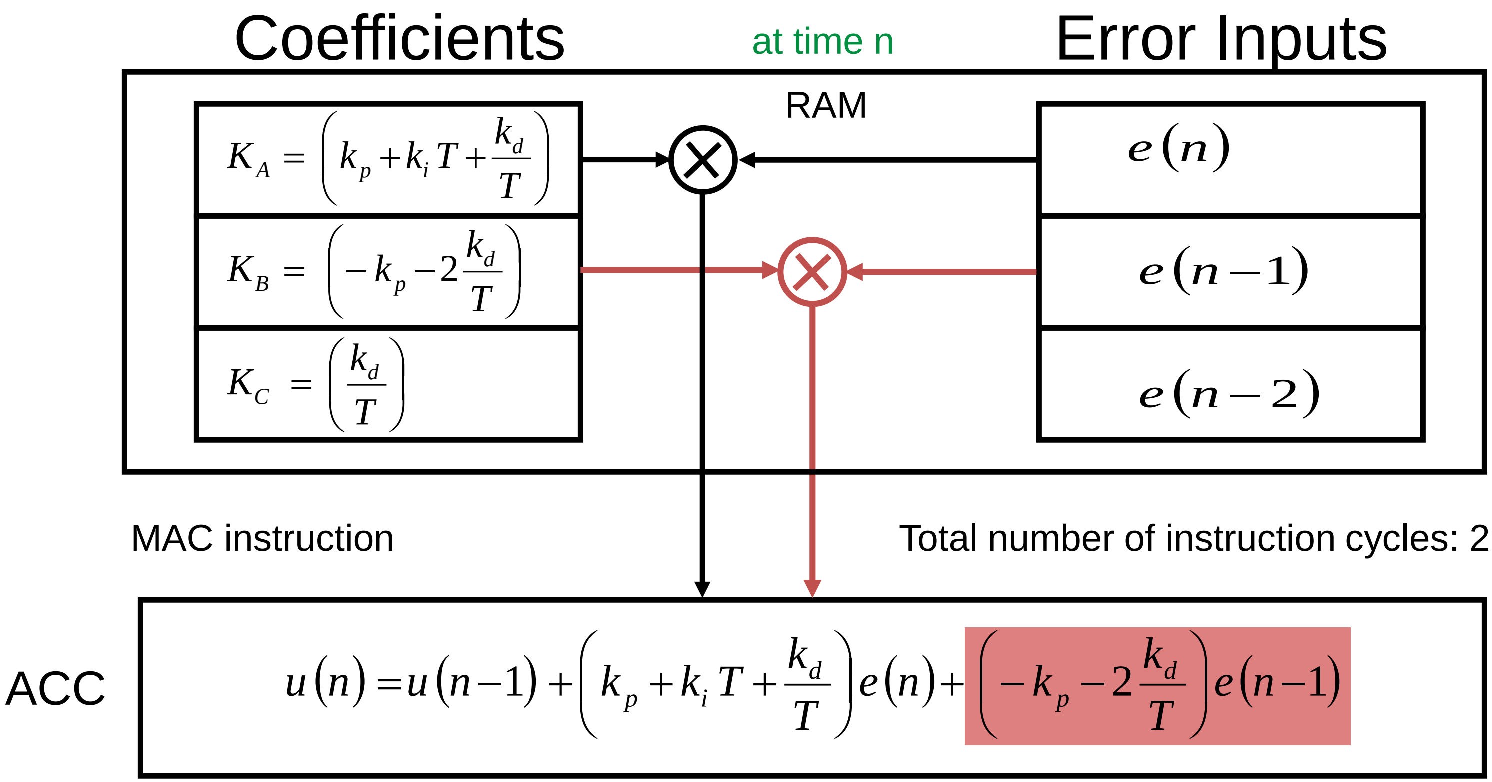
DSC Digital PID

PIDs in DSCs: First Contribution



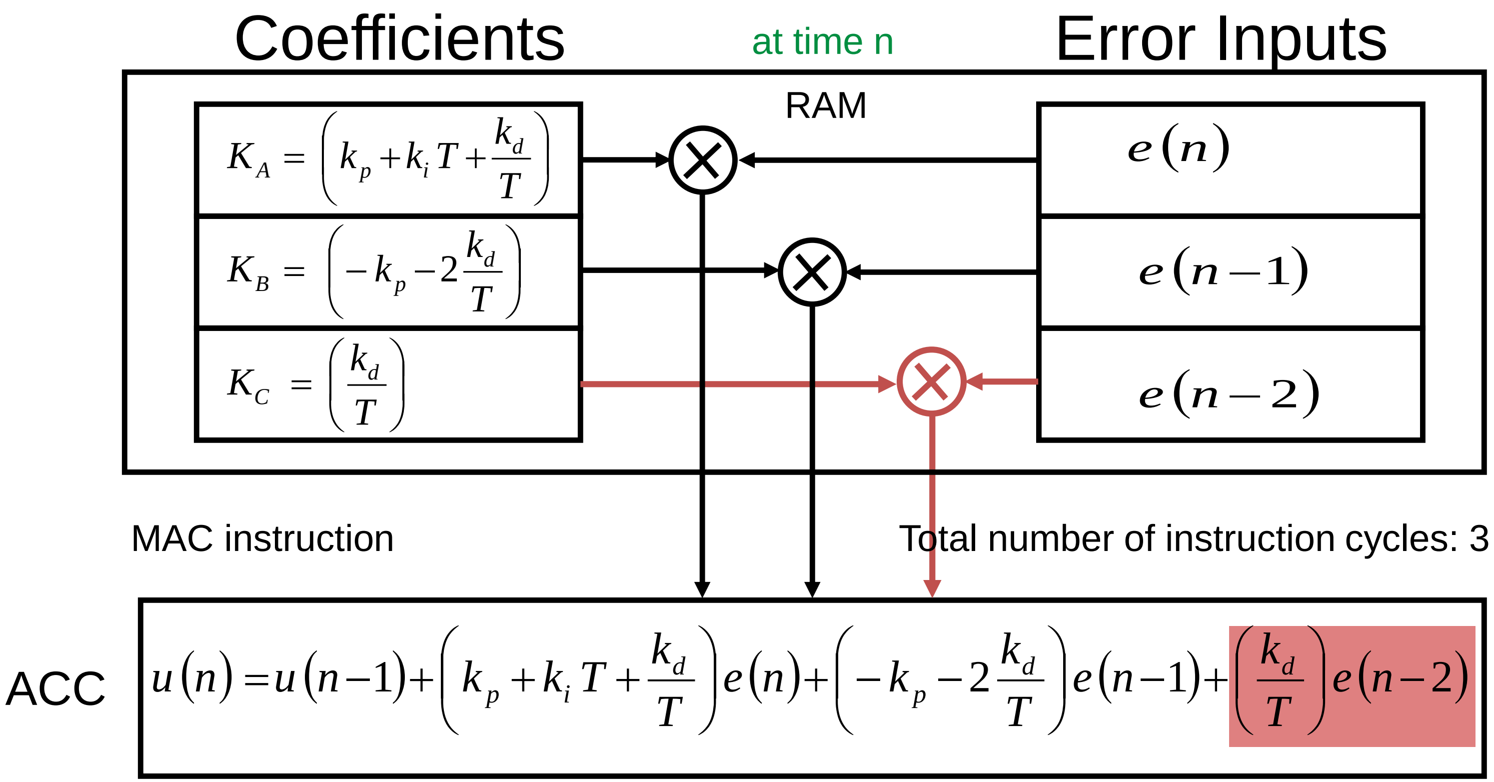
DSC Digital PID

PIDs in DSCs: Second Contribution



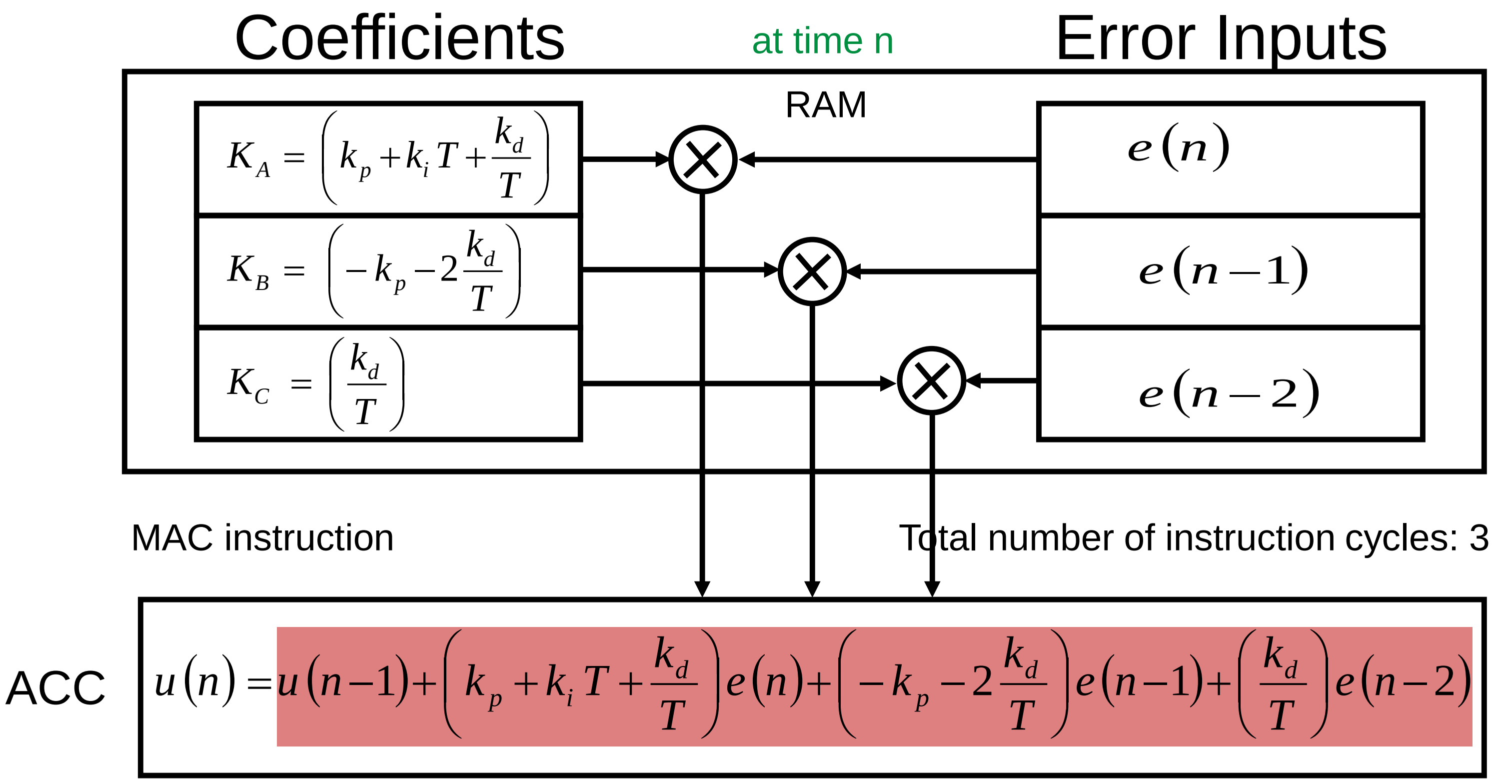
DSC Digital PID

PIDs in DSCs: Third Contribution



DSC Digital PID

PIDs in DSCs: Updated result



Digital PID Equation

DSC Digital PID

PIDs in DSCs: Ready for Next Iteration

Coefficients — at time n + 1 — Error Inputs

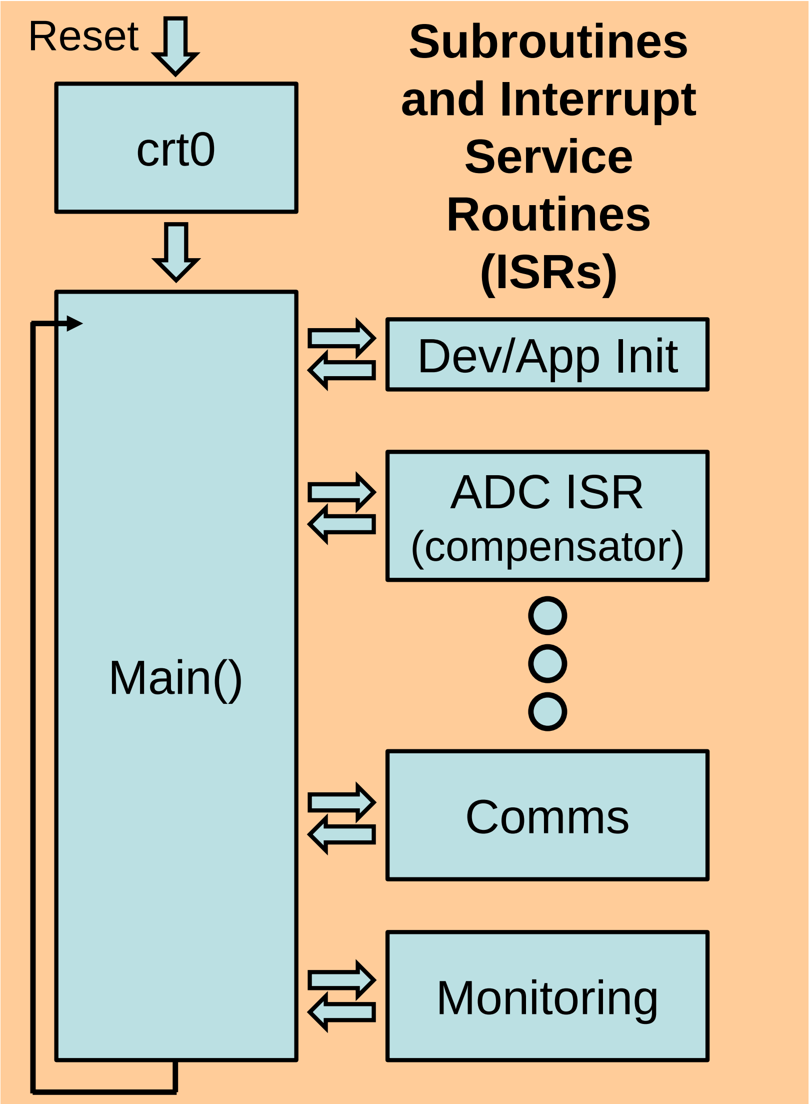
RAM

Coefficients	Error Inputs
$K_A = \left(k_p + k_i T + \frac{k_d}{T}\right)$	empty location
$K_B = \left(-k_p - 2\frac{k_d}{T}\right)$	$e(n)$
$K_C = \left(\frac{k_d}{T}\right)$	$e(n-1)$

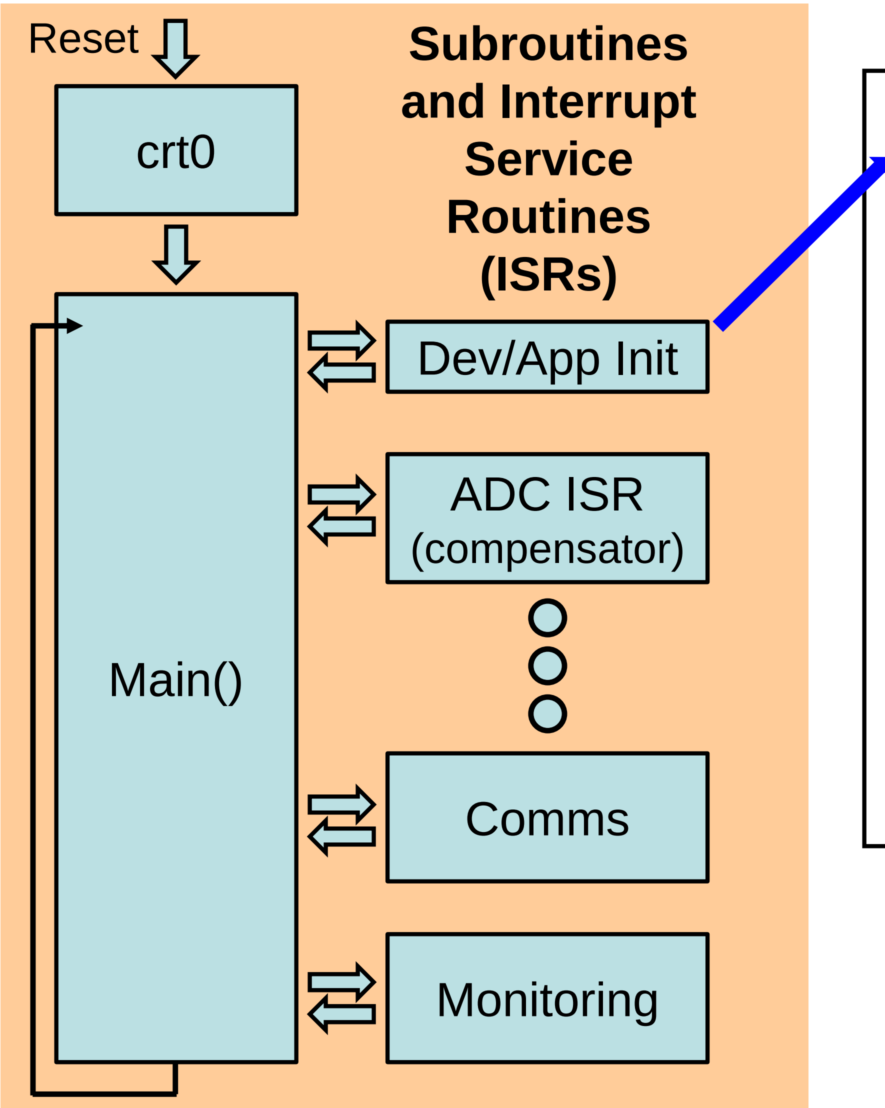
ACC

$$u(n) = u(n-1) + \left(k_p + k_i T + \frac{k_d}{T}\right)e(n) + \left(-k_p - 2\frac{k_d}{T}\right)e(n-1) + \left(\frac{k_d}{T}\right)e(n-2)$$

Typical DSC Firmware Architecture



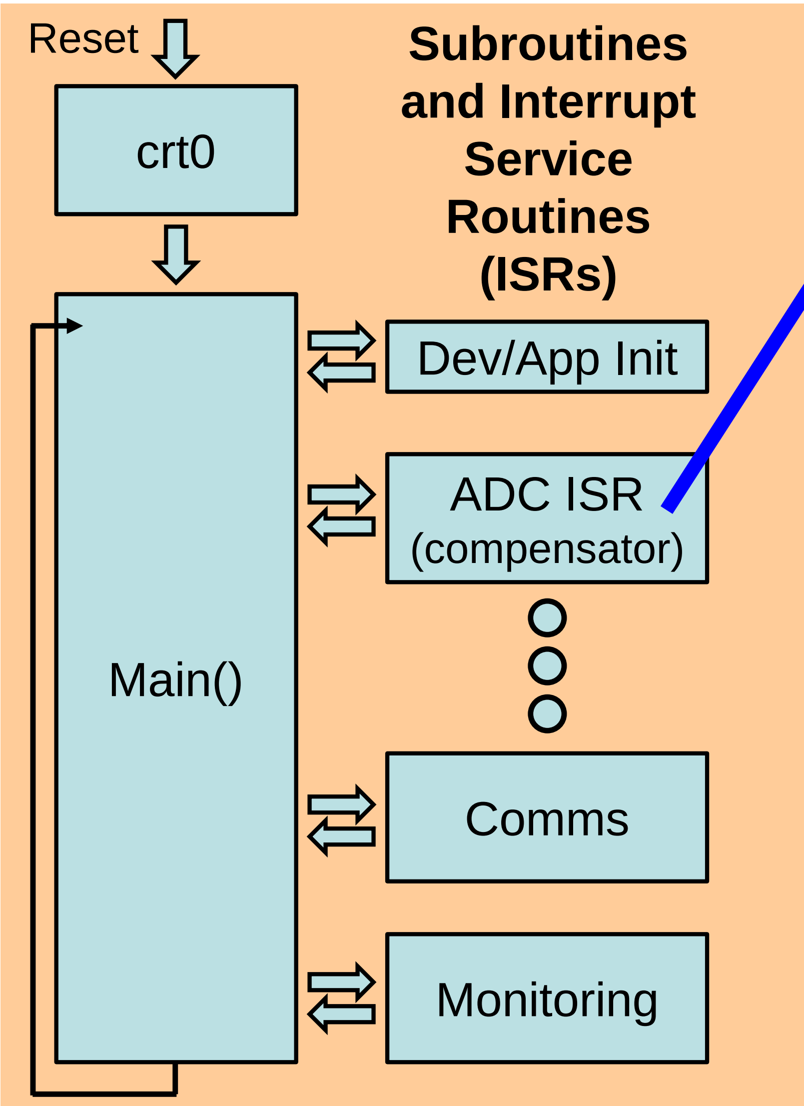
Typical DSC Firmware Architecture



Device / Application Initialization

InitClock();	Setup DSC oscillator and clock
InitComp();	Setup current limit comparator
InitADC();	Setup ADC for output voltage sampling
InitPWM();	Setup PWM
InitIO();	Setup I/O
InitPID();	Initialize compensator, setup pointers, clear error and output history

Typical DSC Firmware Architecture



ADC Interrupt – The Compensator

Clear interrupt flag

Init pointers to coefficients and errors arrays

Read new voltage sample value

Scale the input voltage

Computation of e(n) = Vref – input voltage

Voltage mode PID computation

Update the voltage error history

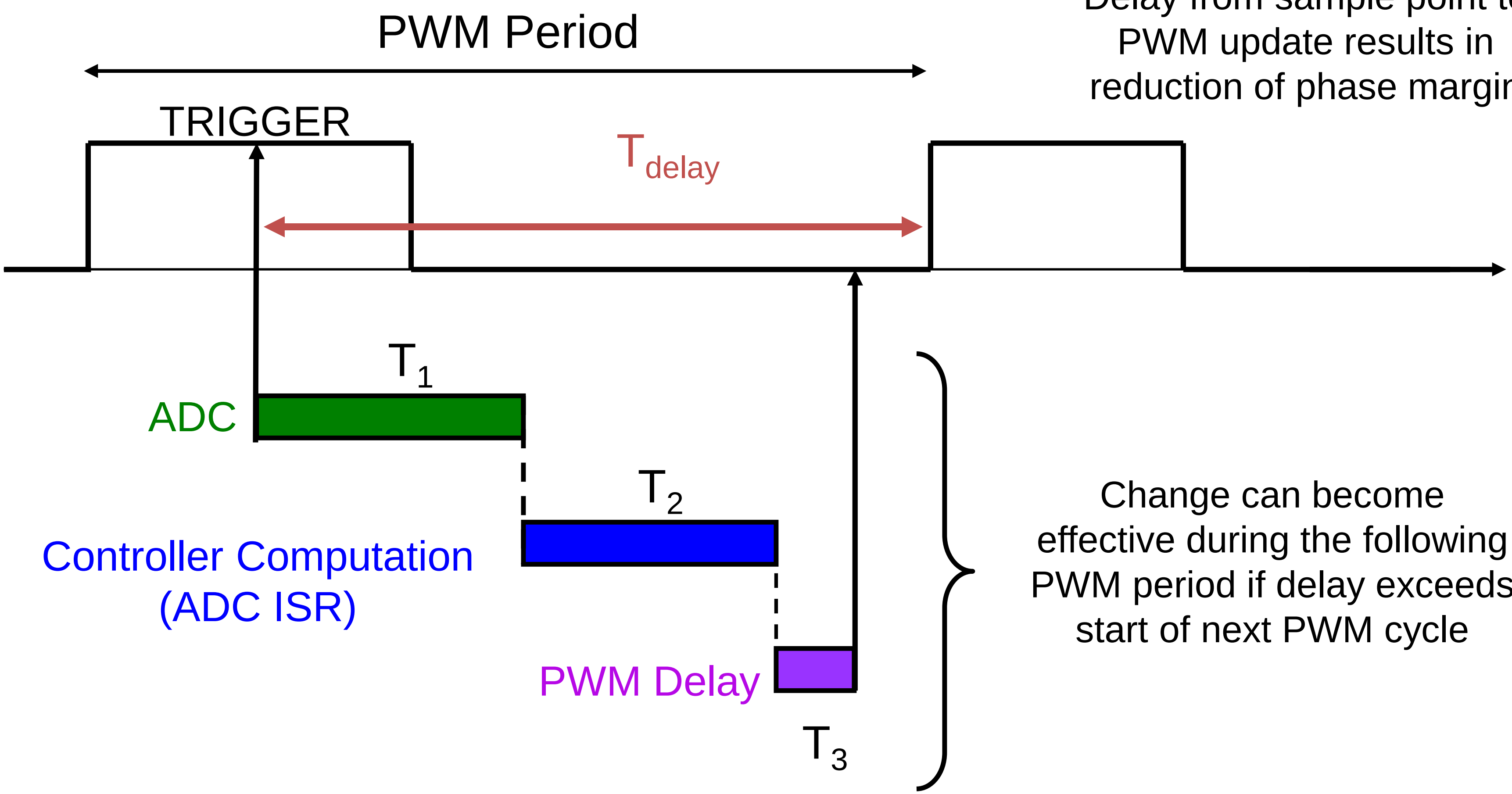
Check if duty cycle is within limits

If needed, clamp to min or max value

Save new PWM duty cycle value in register

Control Loop Timing

Digital Implementation of a Control Loop



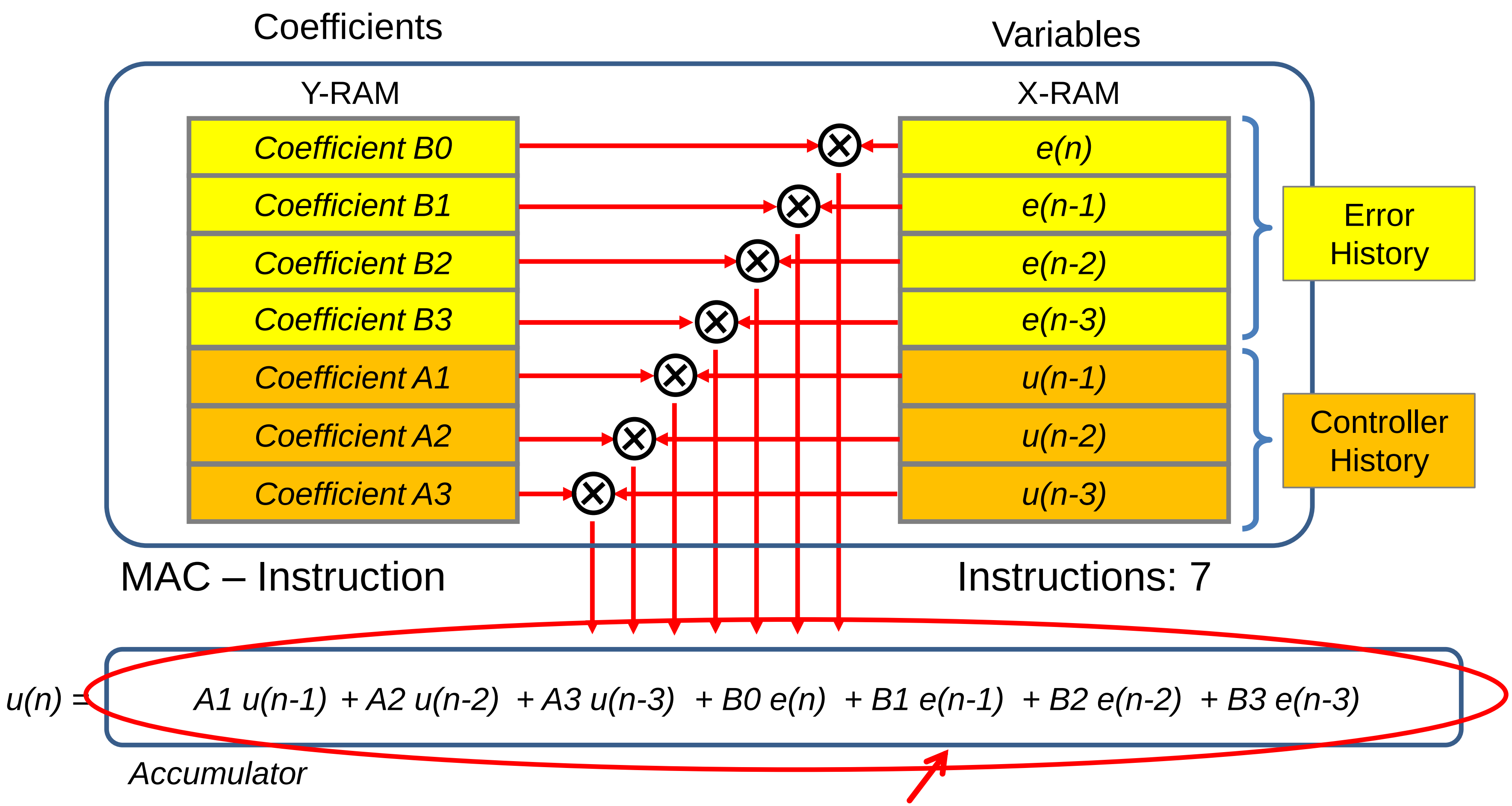
Delay from sample point to PWM update results in reduction of phase margin

Change can become effective during the following PWM period if delay exceeds start of next PWM cycle

Digital Compensator Types

- **Proportional Integral Derivative (PID)**
  - Most common compensator type in industrial control applications, although not ideal for SMPS applications
  - Uses only three coefficients – simple method to find the values
  - Only compensates one pole and one zero of plant
- **Digital 2P2Z (similar to Analog Type II)**
  - Five coefficients – Five MAC instructions to calculate
  - Current-mode converters
- **Digital 3P3Z (similar to Analog Type III)**
  - Seven coefficients – Seven MAC instructions to calculate
  - Voltage-mode buck or boost-derived converters

Linear Difference Equation for 3P3Z Compensator

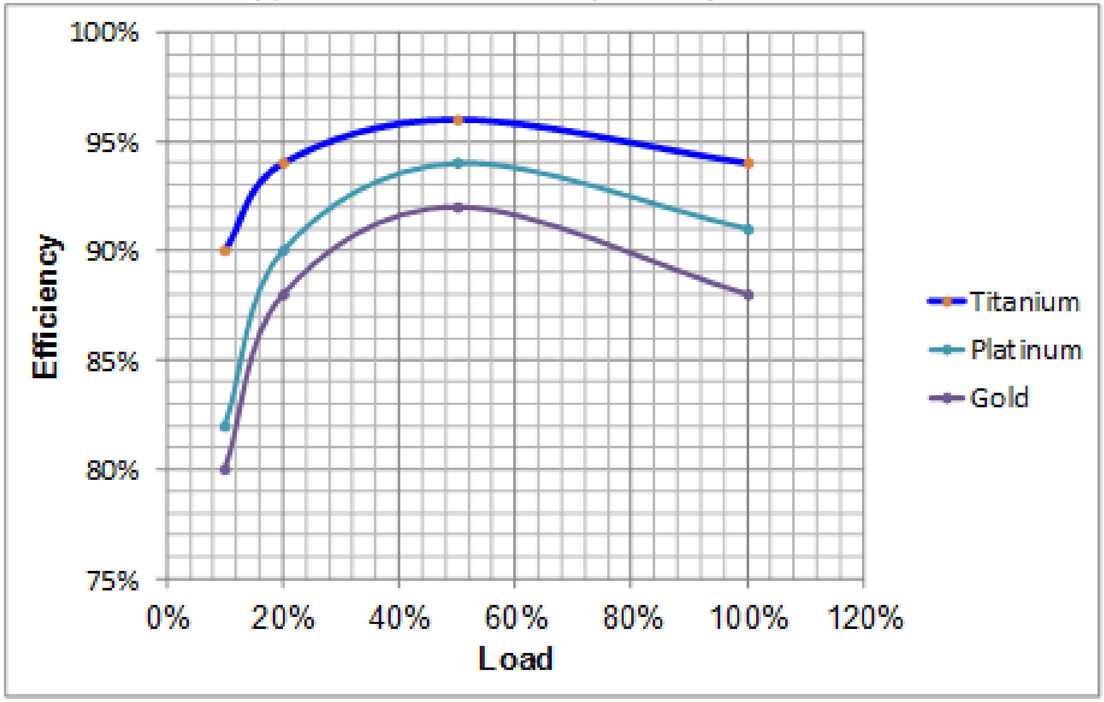


Linear difference equation of a digital type III (3p3z) compensator

Full Digital Control
Adaptive Algorithms

- **Improving Efficiency over widely varying loads**
  - Phase shedding
  - Dead-time adjustment
  - Variable switching frequency
  - Variable bulk voltage

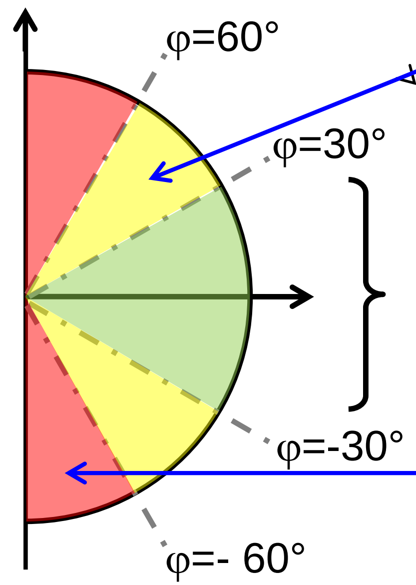
Energy Star Efficiency Requirements



Full Digital Control Predictive and Non-linear Algorithms

- **Dynamic Responsiveness**
  - Non-linear Algorithms
    - Real-time coefficient scaling
  - Predictive Algorithms
    - Bypass damping of control loops

φ represents a vector:
Sum of the absolute value of 3 error samples vs 0 degree vector representing errors averaging zero



Decay / Boost
Modify compensator output by non-linear factor(s)

Steady State
Linear compensator such as 3P3Z

Attack
Override compensator results with min/max values for a limited number of cycles

SMPS Application Notes / Guides

- DS70336 – Buck/Boost Converter PICtail User Guide
- AN1114 – Switch Mode Power Supply Topologies (Part I)
- AN1207 – Switch Mode Power Supply Topologies (Part II)
- AN1106 – Power Factor Correction
- DS70320 – dsPIC SMPS AC-DC Reference Design User Guide
- AN1278 – Interleaved Power Factor Correction
- AN1279 – 1000VA Off-line UPS
- AN1335 – Quarter Brick PSFB DC/DC
- AN1336 – LLC DC/DC
- AN1338 – Grid-Connected Solar Micro-Inverter

SMPS Compensator Libraries

- 3P3Z, 2P2Z, and PID Compensators
  - Good compromise between speed vs. resolution
  - 3P3Z takes ~ 1.6us, 2P2Z takes ~1.4us
- Compensators written to support MATLAB code generation (future)
  - MATLAB needs specific Inputs and Outputs to be defined
  - void SMPS_Controller3P3ZUpdate(SMPS_3P3Z_T* controllerData, volatile uint16_t* controllerInputRegister, int16_t reference, volatile uint16_t* controllerOutputRegister);
    - Pointer to structure (min/max clamps, pre/post shifts, coefficients, and control/error history)
    - Pointer to control input (i.e. ADCBUFx)
    - Control Reference
    - Pointer to control output (PDCx, CMPDACx, etc.)

Microchip / Biricha Digital Power Workshops

Microchip has teamed up with Biricha Digital Power to offer world-leading expertise and training in the field of digital power

- **2015 Workshops:**
  - Feb 10-12: Karlsruhe, Germany (€1400)
  - Jun 2-4: Boston, Massachusetts ($1400)
  - Oct 6-8th: Stockholm, Sweden
  - Oct 27-29th: San Jose, CA

For More Information:

www.microchip.com/biricha

- **Day 1: Introduction to Digital Power Programming**
  - Introduction to dsPIC33 family architecture and peripherals
  - Device setup: Configuration of registers and the oscillator
  - General Purpose: I/O ports, timers and interrupts
  - Digital Power: PWM, comparator and analog to digital converter
  - Introduction to MPLAB® X IDE and program/debug tools
  - Fixed point math and number formatting
  - Setting up the ADC module and using ADC interrupts
- **Day 2: Digital Power Supply Design**
  - Detailed study of analog power supply design fundamentals
  - Step-by-step digital power supply design
  - Implementing a simple digital controller for your digital power supply
  - Stable digital compensators for voltage mode control
- **Day 3: Digital Peak Current Mode Control**
  - Detailed review of analog peak current mode design
  - Sub-harmonic oscillations and slope compensation
  - Digital peak current mode controller design
  - High performance digital power supply design and implementation

Review

- **Review the Levels of Digital Integration**
- **Benefits of Full Digital Control**
- **Digital Controller Basics**
  - Analog SMPS and Digital SMPS Implementations
- **The Digital PID**
  - The Mathematics, Generating the Coefficients, DSC Digital PID Implementation
- **Typical DSC Firmware Architecture**
- **Other Digital Compensator Types**
- **Advanced Digital Control**
  - Adaptive, Non-linear, and Predictive algorithms
- **Additional Resources**
  - App Notes, Libraries, Digital Compensator Design Tool, Workshops

Audience Q & A via Chat

Instructor:
Tom Spohrer,
Product Marketing Manager,
MCU16 Division,
Microchip Technology, Inc.

Moderator:
Rich Nass,
EVP, OpenSystems Media

IF YOU DO NOT SEE THE CHAT MODULE ON YOUR SCREEN,
Click here to join us for the class chat:
http://opsy.st/1QUoTGw

Embedded University

Thanks for joining us

MICROCHIP

Class archive available at:
http://opsy.st/1QUoTGw

E-mail us at:
jgilmore@opensystemsmedia.com

MICROCHIP

Powered by Embedded Computing Design