Full digitally controlled Power Supply design

Olivier Monnier
TI Business Development Manager, C2000 DSP Controllers

Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Implementation
- Next Steps

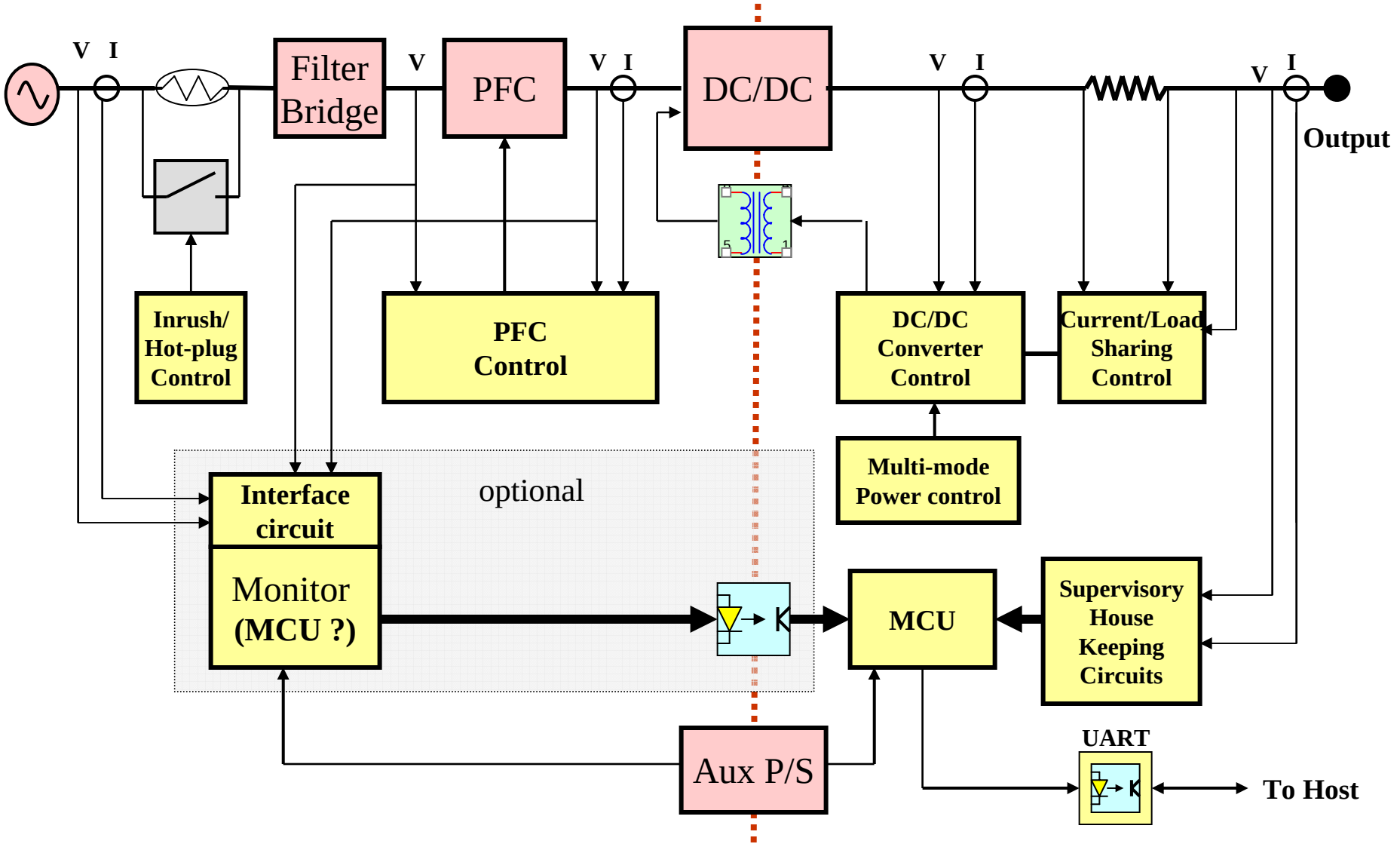
Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Next Steps

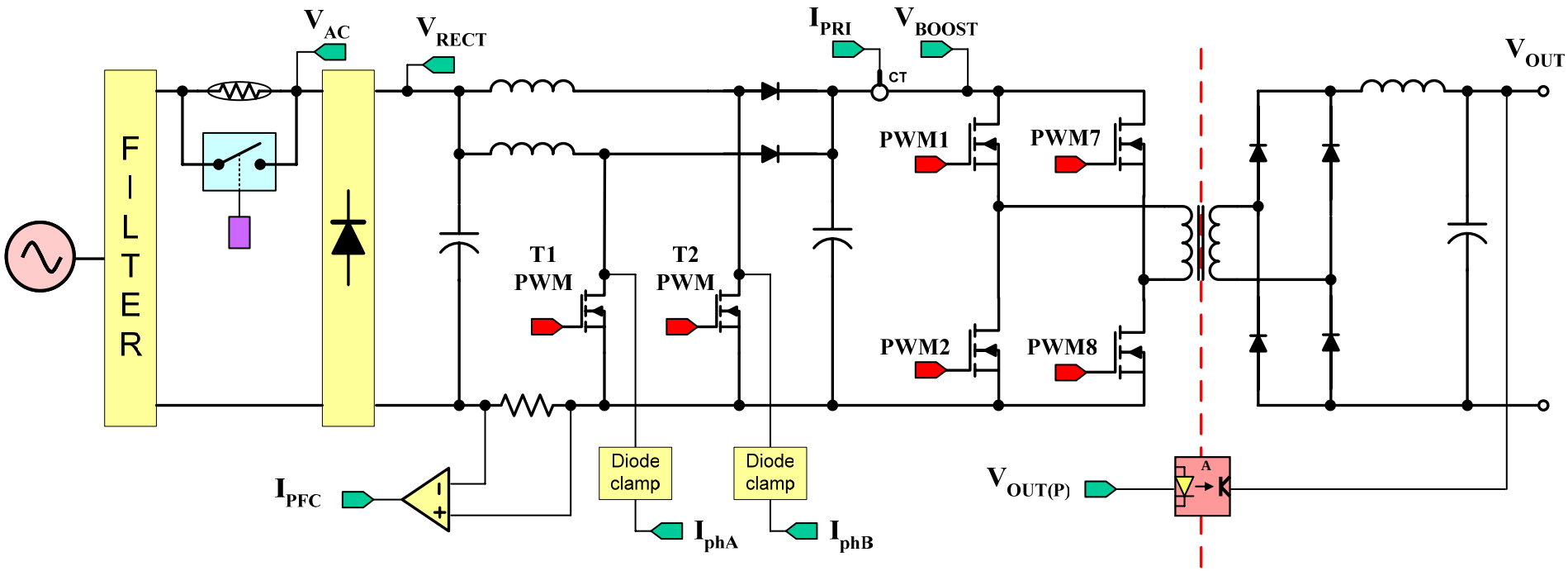
The Digital Vision

- Why Digital Approach for Power Supplies?
- Why DSP Controllers?

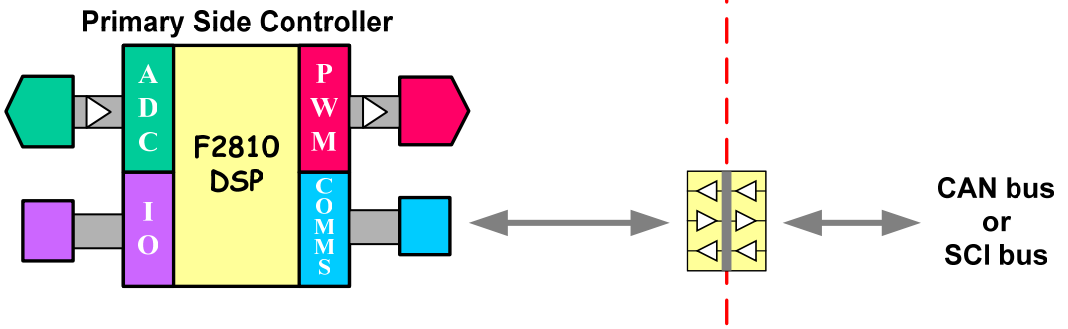
Typical Analog based AC/DC rectifier



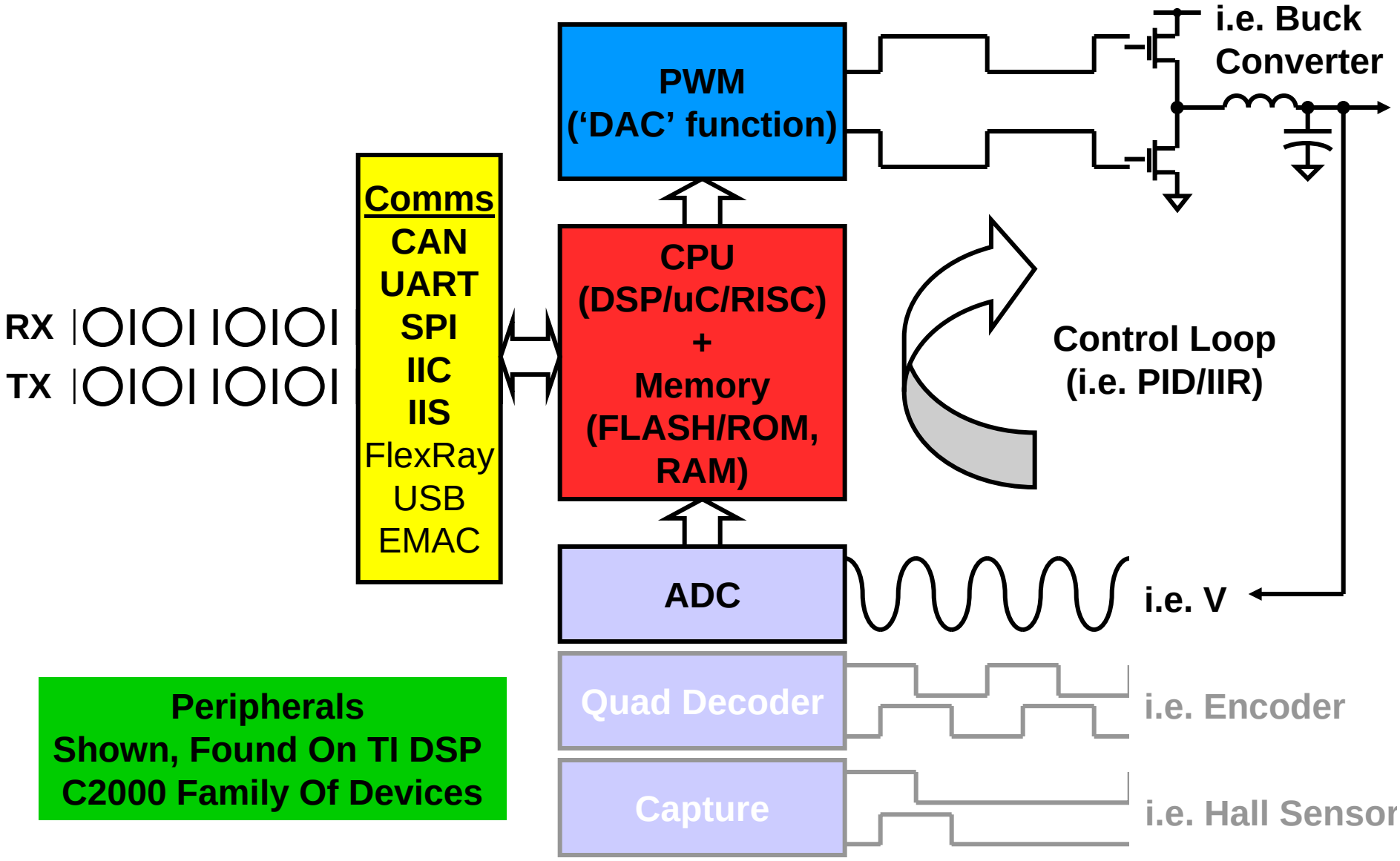
Digital approach with Single Device example for AC/DC Rectifier



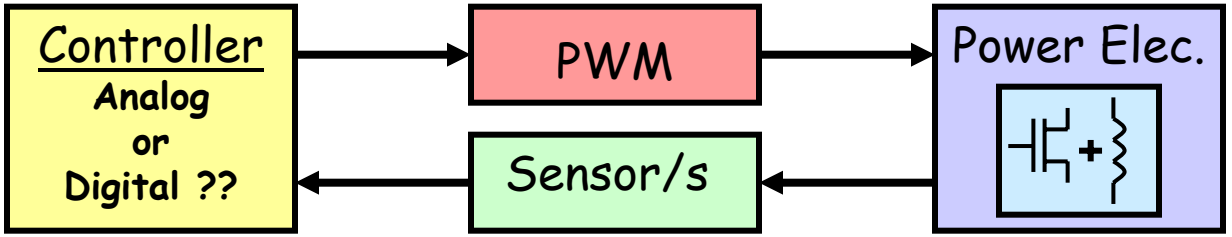
- 1000W / 48 V
- F2810 DSP based
- 2 Phase PFC-IL
- Phase shifted ZVS-FB
- 200 KHz PWM (DC/DC)
- 100 KHz PWM (PFC)



Typical ‘Control’ System On A Chip



Why DSP for Power Supplies?



	Analog Controller	Digital Controller
+	◆ High bandwidth ◆ High resolution ◆ Easy to understand / use ◆ "relatively' low cost ??	◆ Insensitive to environment (temp, drift,...) ◆ High reliability ◆ S/w programmable / flexible solution ◆ Precise / predictable behaviour ◆ Advanced control possible (non-linear, multi-variable) ◆ Can perform multiple loops and "other" functions
—	◆ Component drift and aging / unstable ◆ Component tolerances ◆ Hardwired / not flexible ◆ Limited to classical control theory only ◆ Large parts count for complex systems	◆ Bandwidth limitations (sampling loop) ◆ PWM frequency and resolution limits ◆ Numerical problems (quantisation, rounding,...) ◆ AD / DA boundary (resolution, speed, cost) ◆ CPU performance limitations ◆ System cost

?

DSP Controllers value-Proposition

- Integration
- Flexibility
- Ease of differentiation
- System cost optimization

- Two Main Power Supply domains targeted
  - Industrial power supplies above 1kW
  - Multi-phase DC/DC loops requiring synchronization

Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Next Steps

Controller Considerations for Digital Power Supplies

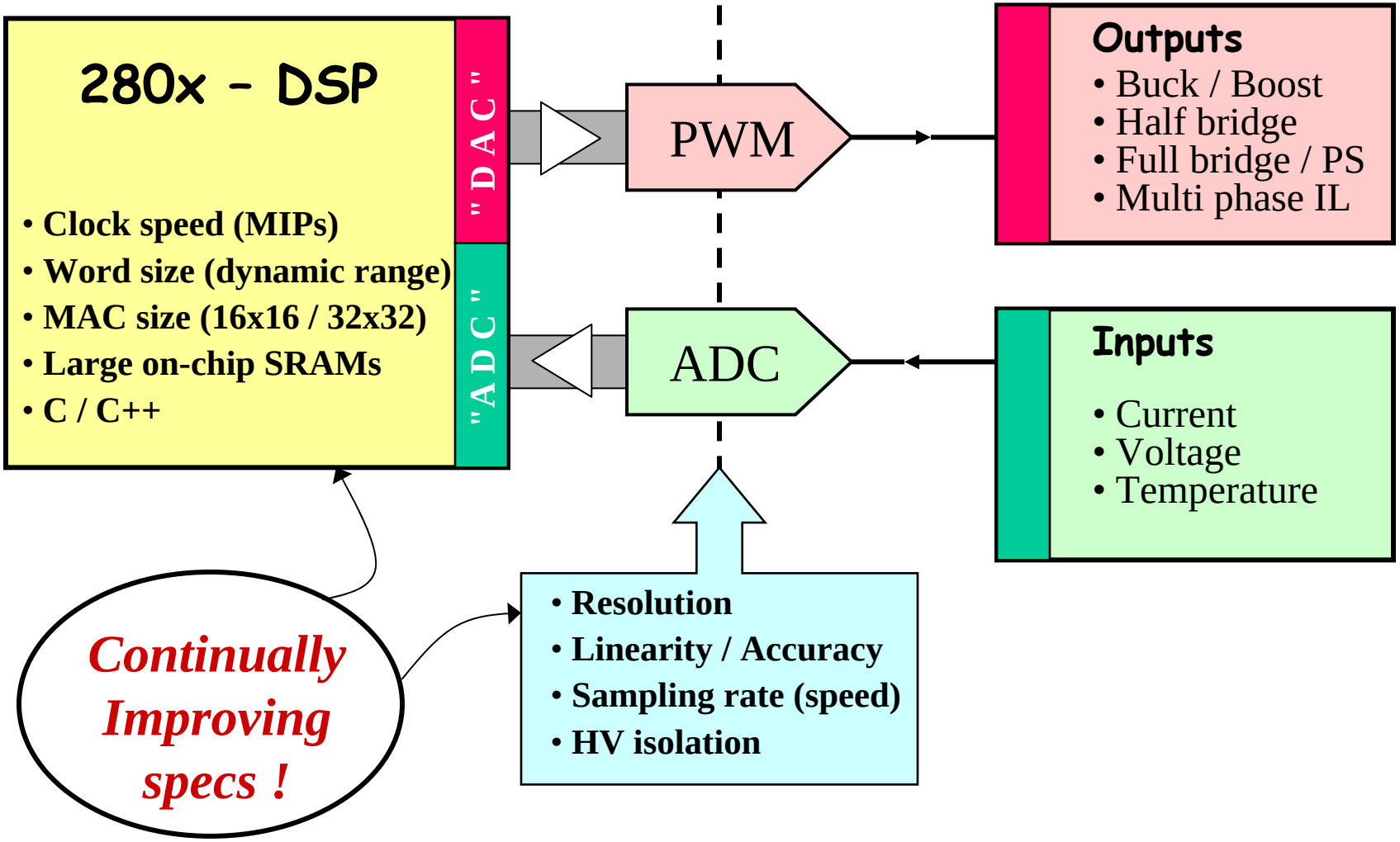
- Ease of Use
- SW
- Reliability
- CPU Performance
- PWM resolution
- Low Interrupt Latency
- Fast Sample Rate
- Numeric Considerations
- Cost
- Technical Support

Fear factor !

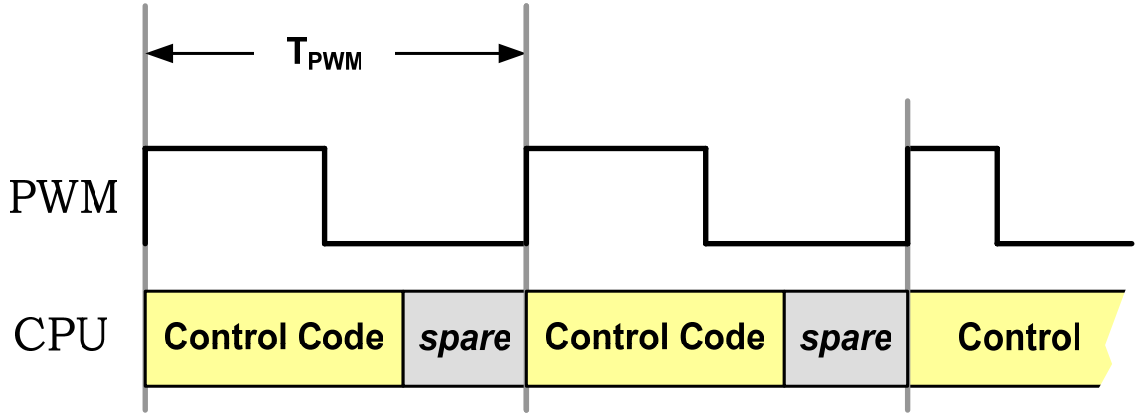
Fully Digital System

- Some Facts, Figures and Capabilities

The Digital Domain……



Processor capability



Inst. vs Algorithm

S/W algorithm	clks
Controller (2 pole / 2 zero)	26
Controller (3 pole / 3 zero)	36
PFC current command	30
PFC OVP	25
BiQuad Filter	46
ZVSFB PWM driver	14
PFC2PHIL PWM driver	26

Instructions vs PWM

PWM freq. (KHz)	PWM per. (uS)	Processor MIPS 40	100	150
50	**20.0**	800	2000	3000
100	**10.0**	400	1000	1500
200	**5.0**	200	500	750
250	**4.0**	160	400	600
300	**3.3**	133	333	500
500	**2.0**	80	200	300
750	**1.3**	53	133	200
1000	**1.0**	40	100	150

MIPS = Million Instruction Per Second

Typical Power Stage Switching Frequencies

Freq. (KHz)	Typ. Application	Power stage
10 ~ 35	Motor Control	3 Phase Inverter
50 ~ 120	UPS	Boost / Buck / ??
80 ~ 160	PFC + boost AC/DC – Rectifier	Single / Multi-phase Interleaved
120 ~ 240	DC/DC (isolated) AC/DC – Rectifier	H-bridge / Full-Bridge / FB-ZVS
200 ~ 1000	DC/DC (non-isol.) DPA-Enterprise	Single phase Buck / Multi-phase Interleaved
1 ~ 4 MHz	DC/DC (non-isol.) DPA / Bricks	Single phase Buck / Multi-phase Interleaved

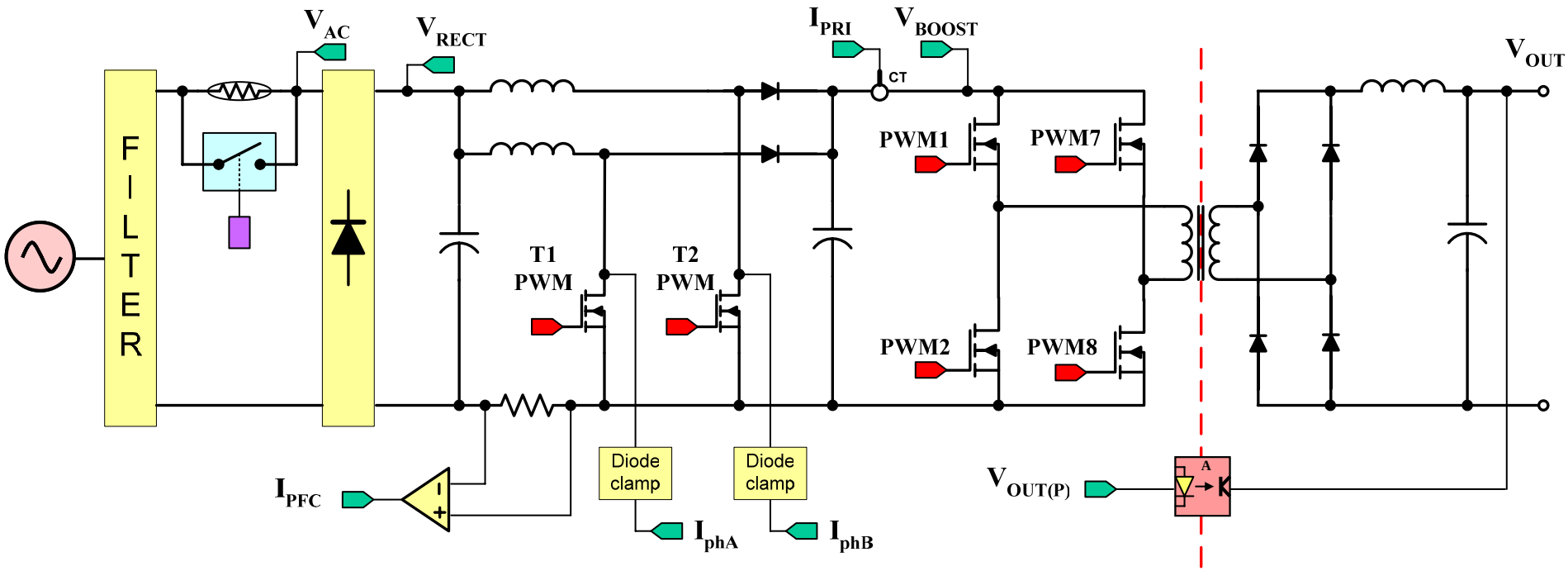
Benefits of higher frequencies

1) Higher power density
2) Smaller magnetics
3) Lighter Power supplies
4) Faster transient response
5) Smaller ripple amplitude

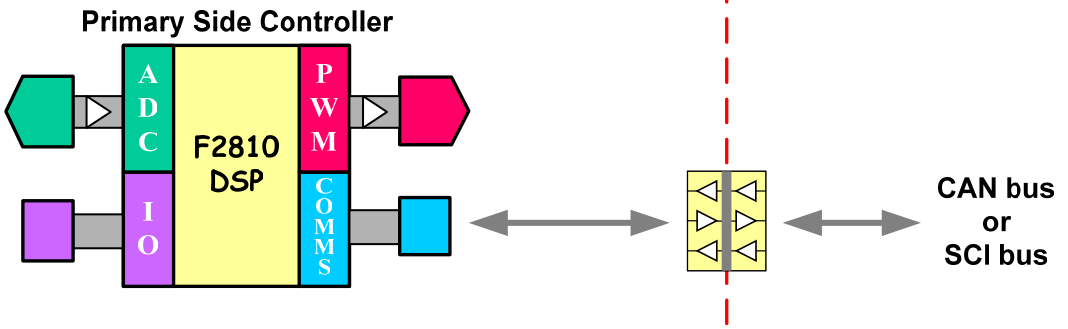
Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Next Steps

Digital approach with Single Device example for AC/DC Rectifier



- 1000W / 48 V
- F2810 DSP based
- 2 Phase PFC-IL
- Phase shifted ZVS-FB
- 200 KHz PWM (DC/DC)
- 100 KHz PWM (PFC)



Digital Control Design Steps

- Choose the topology for each power stage.
- Choose the location for the microprocessor: primary side or secondary side.
- Define the gate drive circuits.
- Define the ADC signal conditioning circuits.
- Choose the configuration of the timing hardware that implements the PWM signals, ADC strobe and interrupt service routine (ISR) timing.
- Architect the firmware: time critical interrupts versus background
- Implement the SW
- Closing the loop digitally offers several advantages when bringing up a system for debug.
- Each stage can be enabled separately.
- Loops can easily be run open-loop, usually by commenting out a line of code.
- Compensation parameters are quickly changed with a few keystrokes.
- Sophisticated diagnostics are possible, such as a circular buffers or complex event triggers.

Digital Control Design Steps

- Choose the topology for each power stage.
- Choose the location for the microprocessor: primary side or secondary side.
- Define the gate drive circuits.
- Define the ADC signal conditioning circuits.
- Choose the configuration of the timing hardware that implements the PWM signals, ADC strobe and interrupt service routine (ISR) timing.
- Architect the firmware: time critical interrupts versus background
- Implement the SW
- Closing the loop digitally offers several advantages when bringing up a system for debug.
- Each stage can be enabled separately.
- Loops can easily be run open-loop, usually by commenting out a line of code.
- Compensation parameters are quickly changed with a few keystrokes.
- Sophisticated diagnostics are possible, such as a circular buffers or complex event triggers.

Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Next Steps

Modularity, re-use efficiency……

Software - is key in Digital Power !

Software playing a more significant role in AC/DC rectifier applications

Traditionally → Analog controlled power stage
S/W role: Supervisory + Monitoring + Comms
8 / 16 bit MCU based

Digital Power → Digitally controlled power stage
S/W role: Supervisory + Monitoring + Comms **+ Closed loop control**
16 / 32 bit DSP based

"Opposing" approaches to Software development

1. "Conservative approach"
- Strictly High level language (e.g. C / C++)
- Conventional function calling / parameter passing
- Real-time OS as needed

→ Wait for release of appropriate device
e.g. 200-300 MIP device @ $5-$10

2. "getting your performance entitlement"
- Combination C / ASM
- "flat" in-line coding
- non-conventional function calling / parameter passing
- simple single ISR structure

→ Push perf. envelope on existing devices
e.g. 100-150 MIP devices @ $5-$10

Defining "GOOD" Software

- Modularity – blocks with well defined inputs / outputs ("cause and effect")
- Multiple instantiation of same module or function
- De-lineation (separation) between code and device peripherals or target h/w i.e. use of peripheral (h/w) drivers
- Re-useable / Re-targetable (maximize return on investment)
- Efficient & high performance – code execution in minimal time
- Easy to use / read / interpret / debug / modify ….. i.e. friendly!

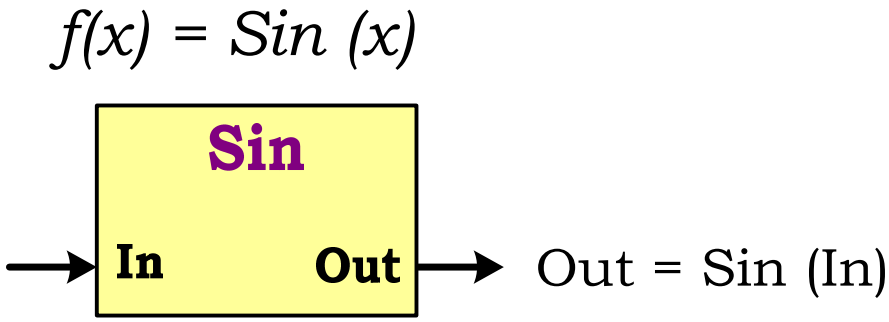
Exploring Modularity

- Function or object with well defined boundaries
- Clear relationship between inputs / outputs ("cause / effect")
- Used multiple times, while maintaining a single source

  "Multiple Instantiation"
- Re-entrant (i.e. supports "nesting of itself")
- "trust for now, explore / understand later"

Module example 1

- Single In / Single out
- Non-configurable
- No History
- Multiple Instantiation

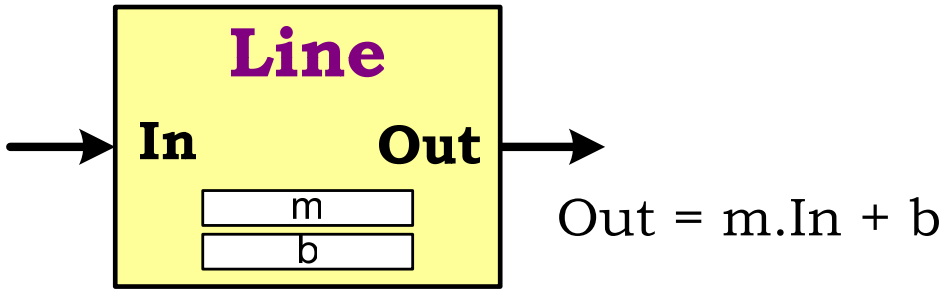


Exploring Modularity

Module example 2

- Single In / Single out
- Configurable
- m, b, Constant ? or Variable ?
- No History
- Multiple Instantiation

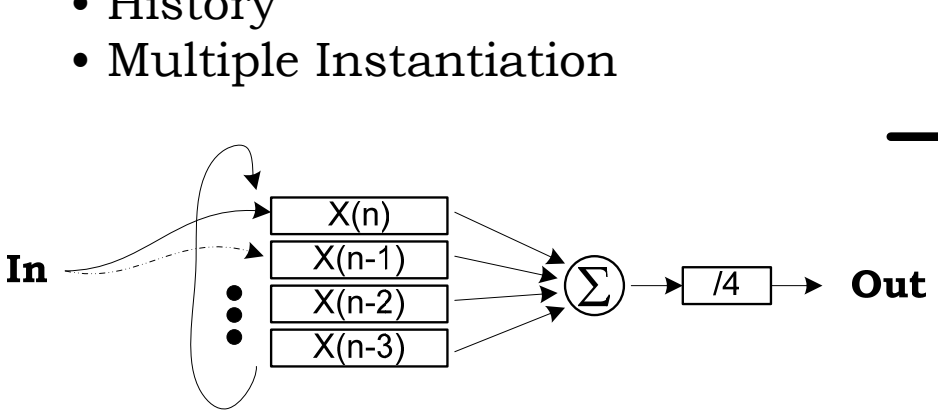
$$f(x) = mx + b$$

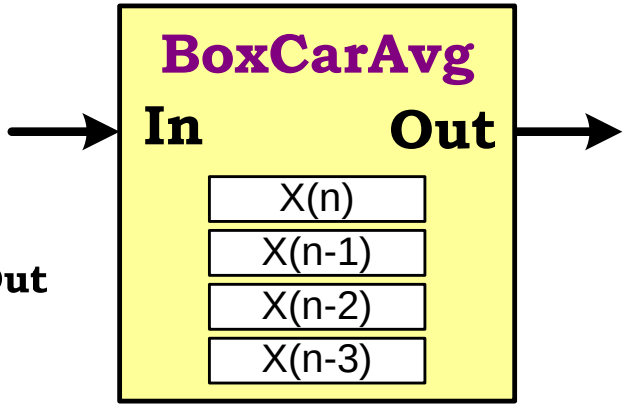


Module example 3

- Single In / Single out
- Non-Configurable
- History
- Multiple Instantiation

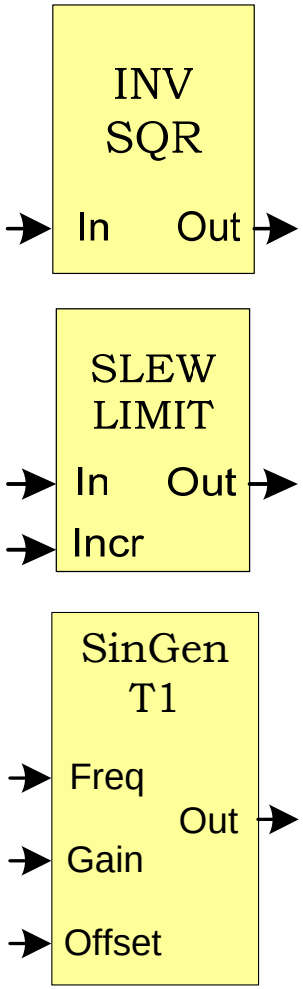
$$f(x) = (x_n + x_{n-1} + x_{n-2} + x_{n-3}) / 4$$



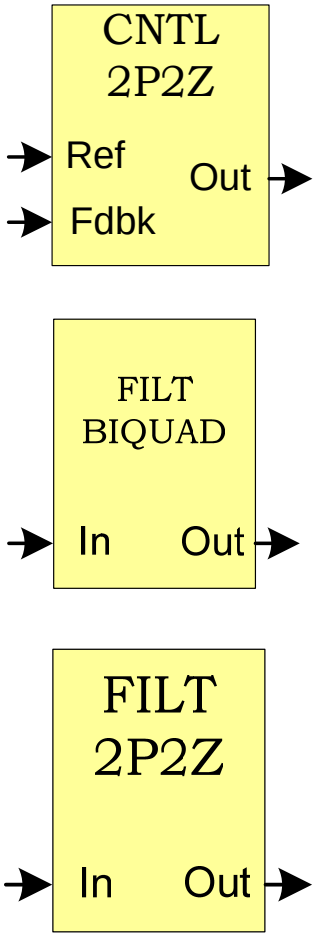


Module Types

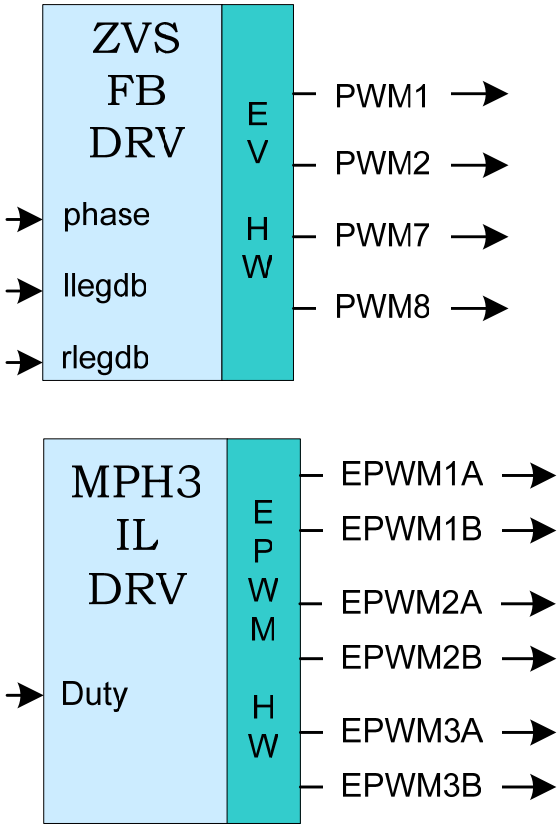
Application Indep. / Peripheral Indep.



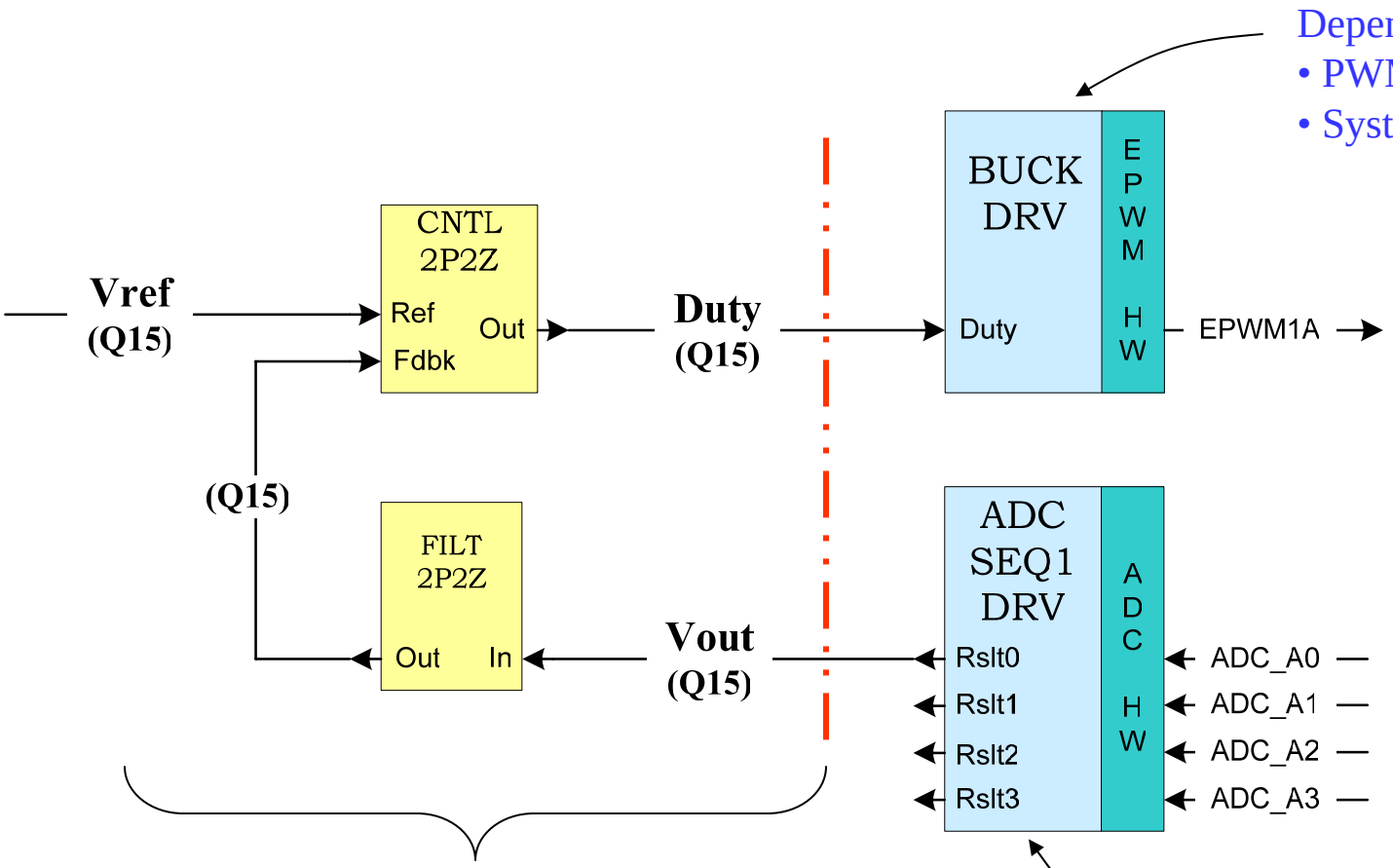
Application Config. / Peripheral Indep.



Application Config. / Peripheral Depend. ("**Peripheral Driver**")



Peripheral Drivers



Q-Math Representation

Fixed point format – S I . F (Sign / Integer . Fraction)

Q15	**S.FFF FFFF FFFF FFFF**	**- 1 < N < +0.99999…**
Q14	**SI.FF FFFF FFFF FFFF**	**- 2 < N < +1.99999…**
Q13	**SII.F FFFF FFFF FFFF**	**- 4 < N < +3.99999…**
Q12	**SIII. FFFF FFFF FFFF**	**- 8 < N < +7.99999…**
Q0	**SIII IIII IIII IIII**	**- 32,768 < N <+32,767**

Qn x Qm = Qn+m, e.g. Q15 x Q14 = Q29

SSI.F FFFF FFFF FFFF FFFF FFFF FFFF FFFF (32 bit format)

SI.FF FFFF FFFF FFFF (adjusted for Q14, 16 bit format)

e.g. Q15 x Q15 = Q30

SS.FF FFFF FFFF FFFF FFFF FFFF FFFF FFFF (32 bit format)

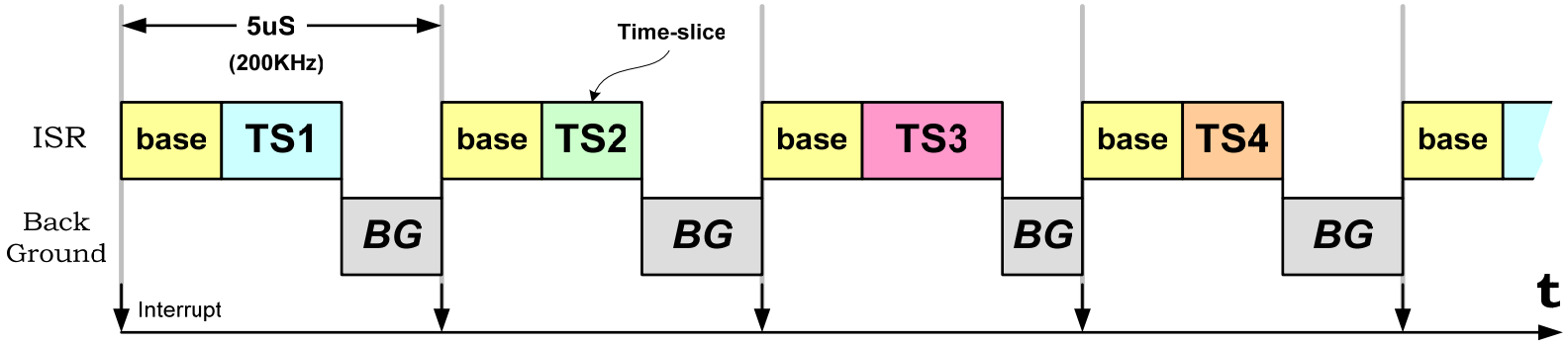
S.FFF FFFF FFFF FFFF (adjusted for Q15, 16 bit format)

Exploring Ideas / Methods for "Good" software

1. System Framework – Background loop ("C") + one ISR ("ASM") with Time-Slicing for control loop
2. In-line coding for the ISR
3. Assembly Macros for the control loop modules
4. Indirect or "pointer based" parameter passing / data flow
5. "Signal Net" based Module connectivity → for the analog guys!

Exploring Ideas / Methods

1. System Framework



- Good choice for addressing many power systems (even complex ones)
- Simple to use and understand
- Efficient (incurs only 1 ISR context save/restore)
- Deterministic (all events synchronous and submultiples of ISR freq.)
- High degree of visibility during debug and development

Back-ground loop (BG)

- C / C++, large code, complex, feature rich, key customer differentiator
- System intelligence / personality, heavy in “if then else”

Interrupt Service Routine (ISR) – Main control loop

- “lean and mean” in-line assembly (ASM) results in a very small footprint.
- Typically “Math function” type code (very few “if then else” branches or loops)
- Once developed, changes very little. Low maintenance burden.

Exploring Ideas / Methods

2. In-line assembly ISR

Fear factor !

How complex ?

How much code development ?

How much maintenance burden ?

How wasteful on memory ?

Number of Instructions / cycles (words)

PWM	MIPS	
(KHz)	100	150
200	500	750
250	400	600
300	333	500
350	286	429
400	250	375
500	200	300

3. ASM Macros – great for modularity !

Modern compilers support:

- Macro parameter passing
- Macro variable & lable substitution

Benefits:

- No call/return overhead (save 8 cycles/call)
- Can easily build self contained modules (modular!)
- Supports multiple instantiation
- Supports “Re-entrancy”
- Re-useable

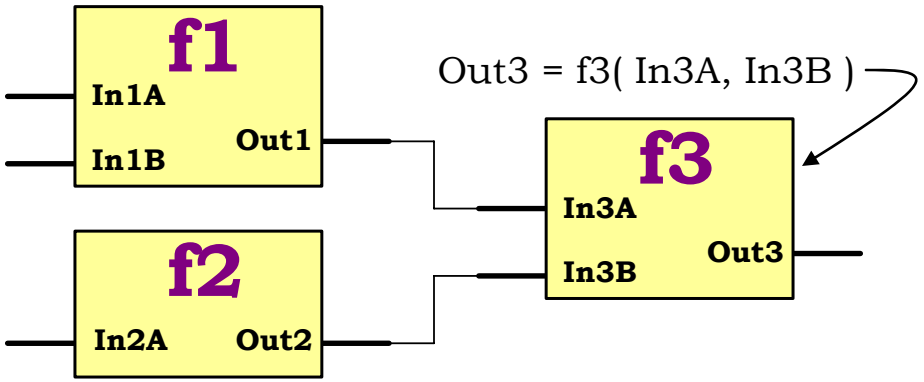
% impact per 10 instructions

PWM	MIPS	
(KHz)	100	150
200	2.0%	1.3%
250	2.5%	1.7%
300	3.0%	2.0%
350	3.5%	2.3%
400	4.0%	2.7%
500	5.0%	3.3%

Exploring Ideas / Methods

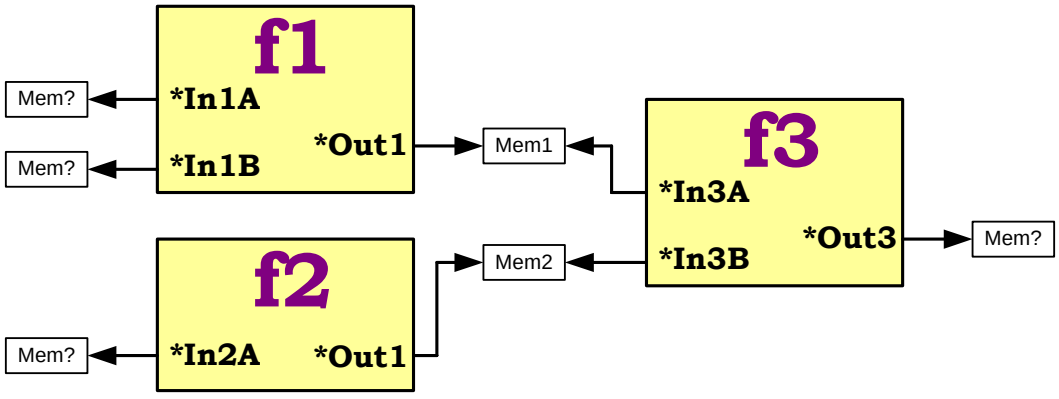
4. Pointer based parameter passing (data flow)

Conventional approach



Pseudo code	*Overhead*
move ? , In1A	(2)
move ? , In1B	(2)
call **f1**	(8)
move ? , In2A	(2)
call **f2**	(8)
move Out1 , In3A	(2)
move Out2 , In3B	(2)
call **f3**	(8)
move Out3 , ?	(2)

Pointer based approach



Pseudo code (without macros)

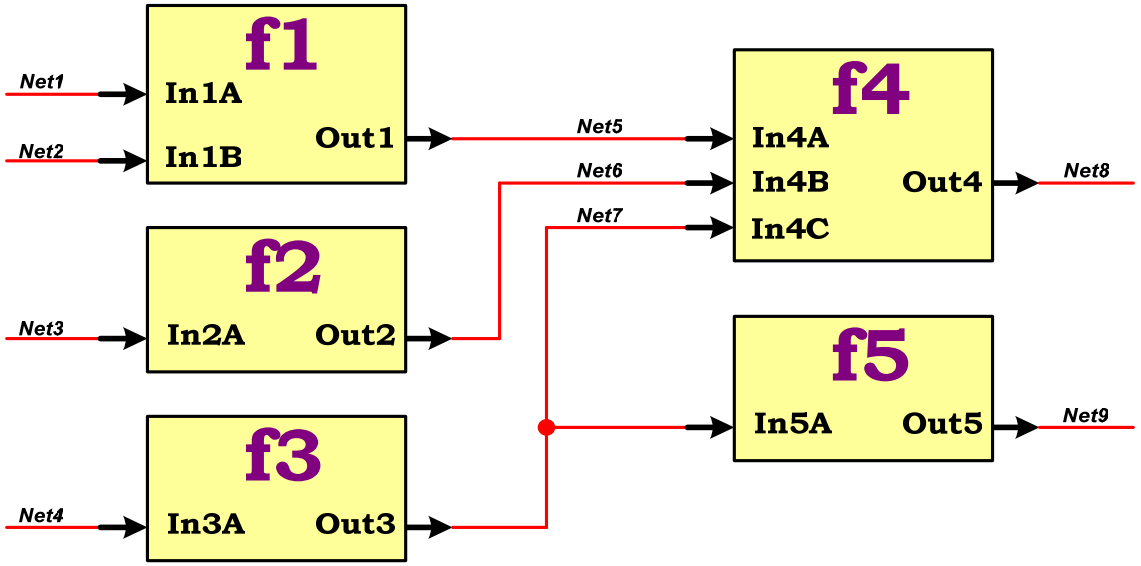
call **f1**	(8)
call **f2**	(8)
call **f3**	(8)

Pseudo code (with macros)

call **f1**	**(zero)**
call **f2**	**(zero)**
call **f3**	**(zero)**

Exploring Ideas / Methods

5. "Signal Net" based module connectivity



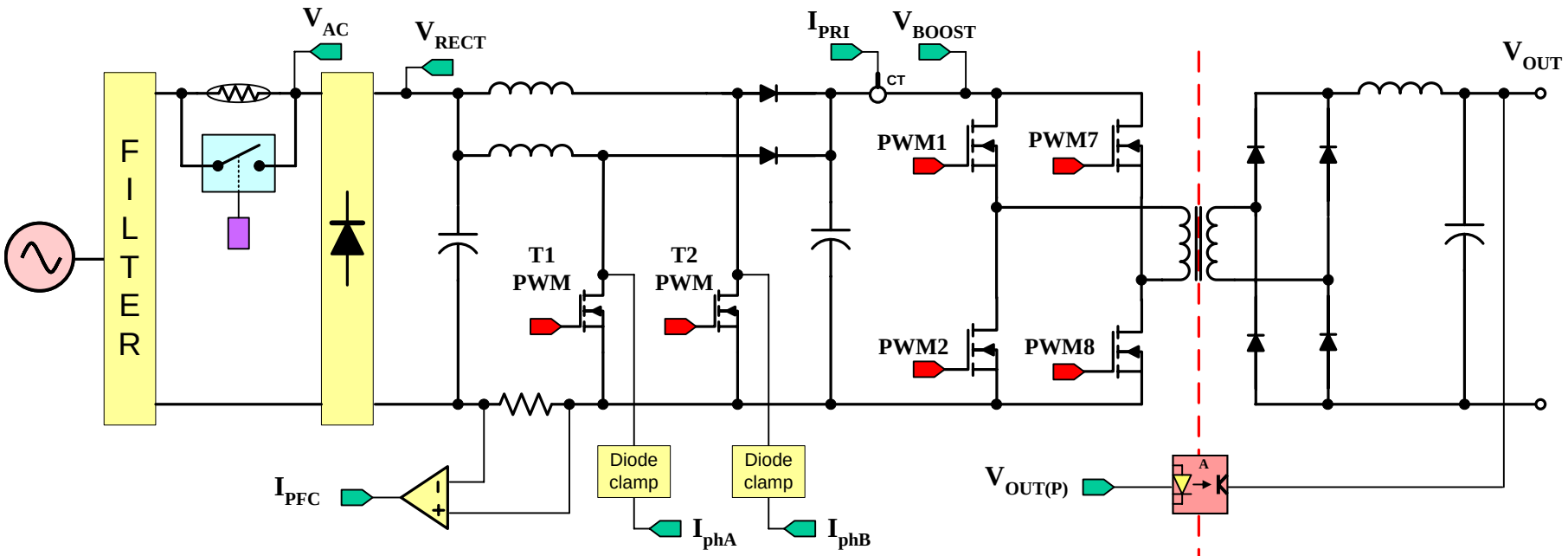
Initialization time ("C")

```
// pointer & Net declarations
Int *In1A, *In1B, *Out1, *In2A,...
Int Net1, Net2, Net3, Net4,...

// "connect" the modules
In1A=&Net1; In1B=&Net2; Out1=&Net5;
In2A=&Net3; Out2=&Net6;
In3A=&Net4; Out3=&Net7;
In4A=&Net5; In4B=&Net6; In4C=&Net7; Out4=&Net8;
In5A=&Net7; Out5=&Net9;
```

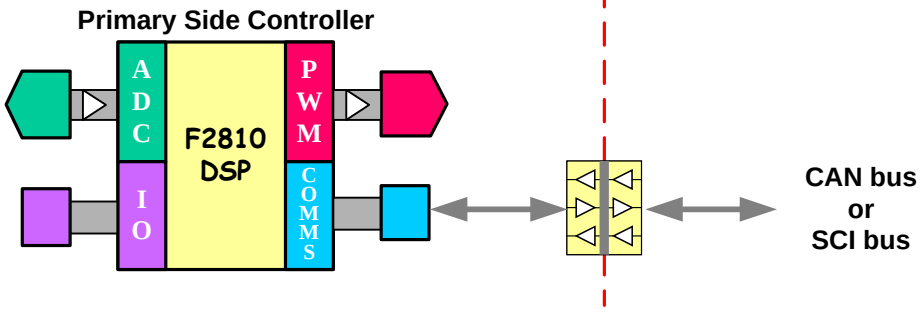
Run time (ASM macros)

```
; Execute the code

        f1
        f2
        f3
        f4
        f5
```

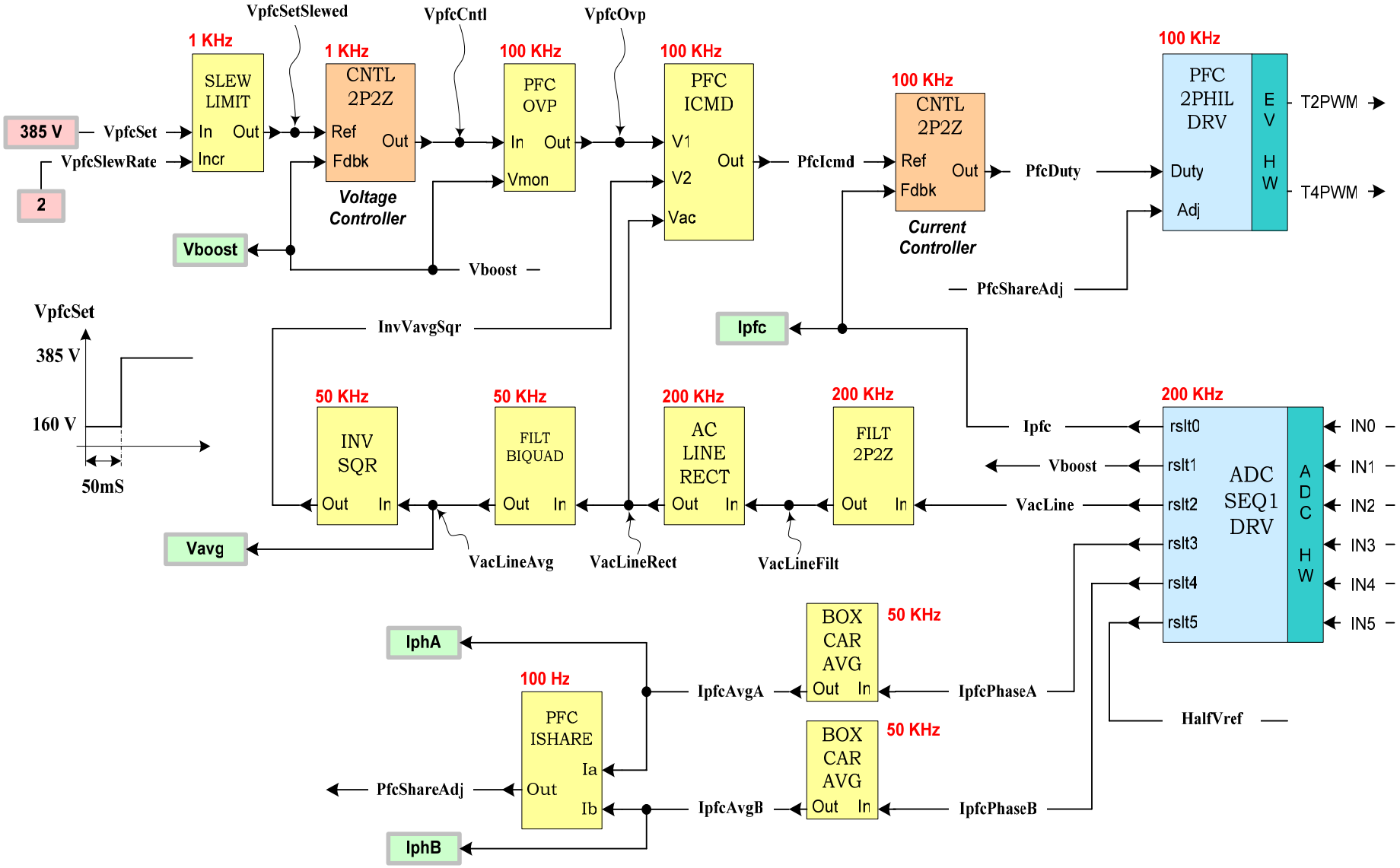
Digitally controlled AC/DC rectifier – an example



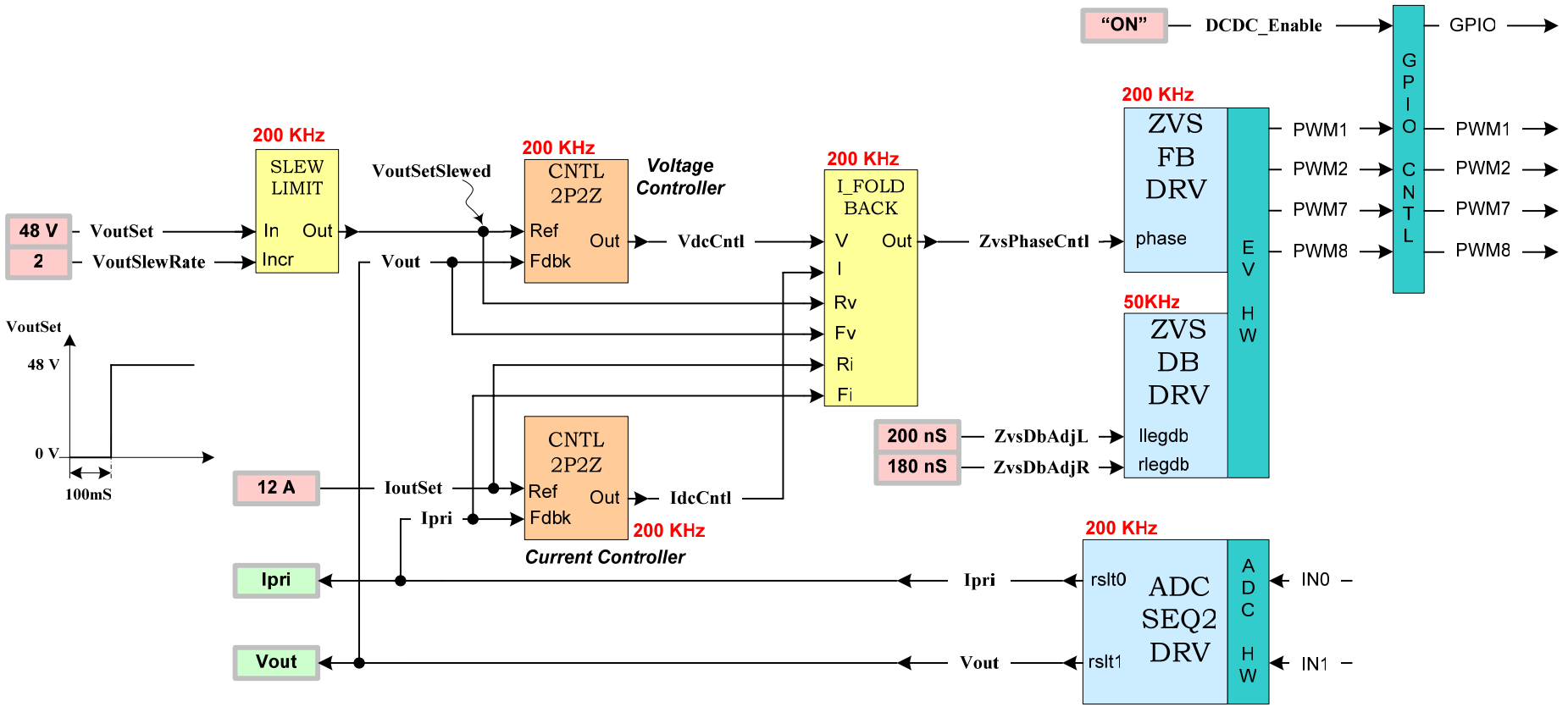
- 1000W
- F2810 DSP based
- 2 phase interleaved PFC
- Phase shifted ZVS-FB
- 200 KHz PWM (DC/DC)
- 100 KHz PWM (PFC)



PFC (2PHIL) Software control flow



DC-DC (ZVSFB) Software control flow

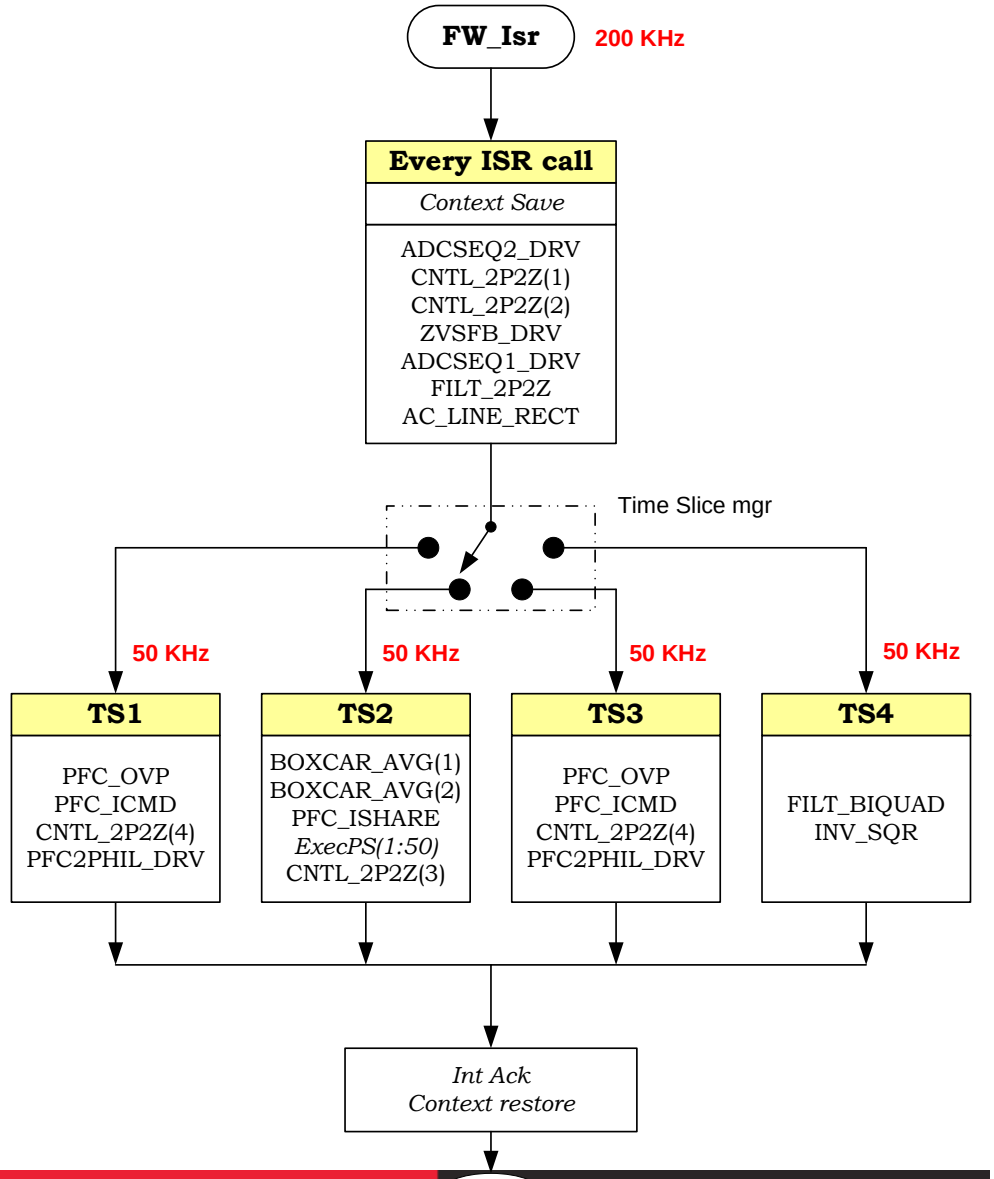


CPU Bandwidth utilization

MIPS =	**100**	# inst / uS =	100	PWM(KHz) =	**200**
# TS =	4	# inst / time slice =	500	PWM(bits) =	9.0
S. rate =	**200**	Sampling period =	5.0		

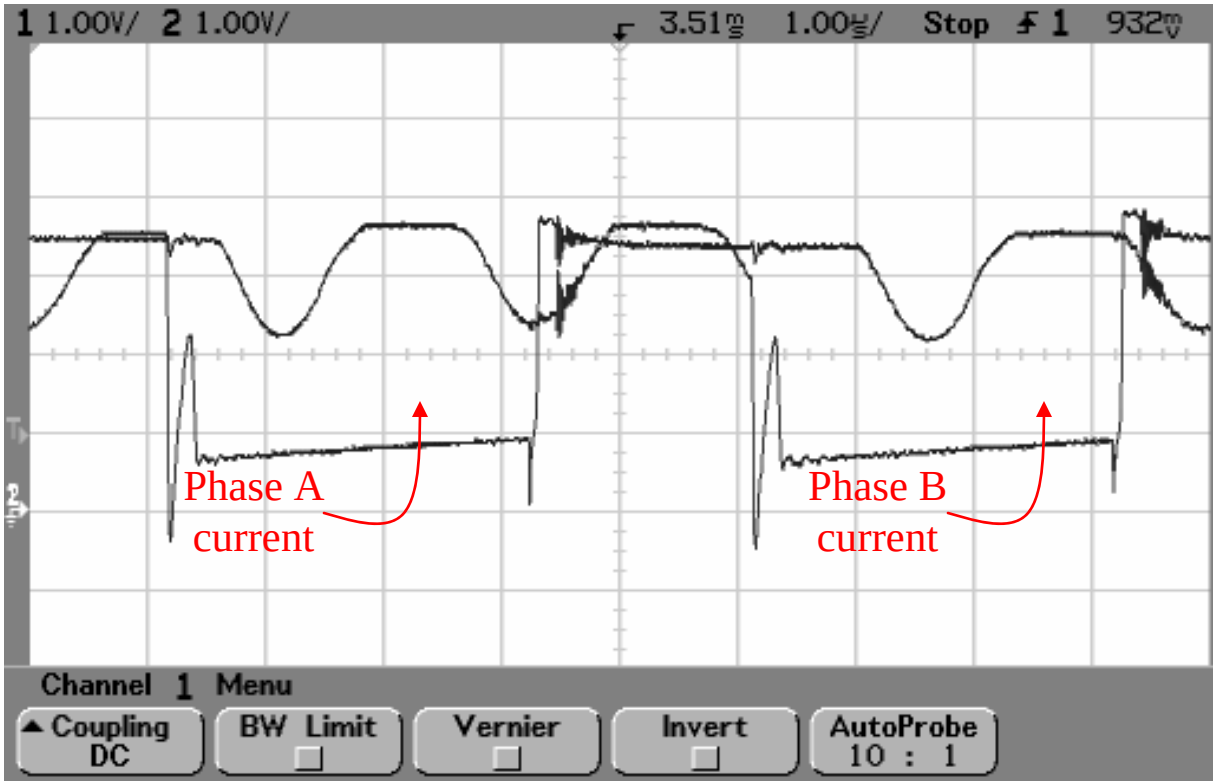
ISR	Rate	Function / Activity	# Cyc	Tot. Cyc.	Stats
All	200KHz	Context Save / Restore	32	**292**	**%Util**
	200KHz	ISR Call / Return / Ack	24		**58%**
	200KHz	Time slice Mgmt	12		
	200KHz	**ADCSEQ2_DRV**	14		
	200KHz	**CNTL_2P2Z 1 (V loop)**	36		
	200KHz	**CNTL_2P2Z 2 (I loop)**	36		
	200KHz	**I_FOLD_BACK**	25		
	200KHz	**ZVSFB_DRV**	14		
	200KHz	**ADCSEQ1_DRV**	57		
	200KHz	**FILT_2P2Z**	35		
	200KHz	**AC_LINE_RECT**	7		
TS1	100KHz	**PFC_OVP**	25	**117**	**%Util**
	100KHz	**PFC_ICMD**	30		**82%**
	100KHz	**CNTL_2P2Z 4 (I loop)**	36		**#Cyc. Rem.**
	100KHz	**PFC2PHIL_DRV**	26		**91**
TS2	50KHz	**BOXCAR_AVG 1**	42	**145**	**%Util**
	50KHz	**BOXCAR_AVG 2**	42		**87%**
	100 Hz	**PFC_ISHARE**	15		**#Cyc. Rem.**
	50KHz	Execution Pre-scaler(1:50)	10		**63**
	1KHz	**CNTL_2P2Z 3 (V loop)**	36		
TS3	100KHz	**PFC_OVP**	25	**117**	**%Util**
	100KHz	**PFC_ICMD**	30		**82%**
	100KHz	**CNTL_2P2Z 4 (I loop)**	36		**#Cyc. Rem.**
	100KHz	**PFC2PHIL_DRV**	26		**91**
TS4	50KHz	**FILT_BIQUAD**	46	**124**	**%Util**
	50KHz	**INV_SQR**	78		**83%**
					#Cyc. Rem.
					84

BG		Function / Activity	# inst.	Tot.Cyc.	Stats
		Comms + Supervisory	400	**434**	
		+ Soft-Start + Other ?			
		SLEW_LIMIT 1	17		
		SLEW_LIMIT 2	17		
			% ISR utilization =	**87%**	
			Spare ISR MIPS =	**12.6**	
			BG loop rate (KHz) / (uS) =	**29.0**	**34.4**



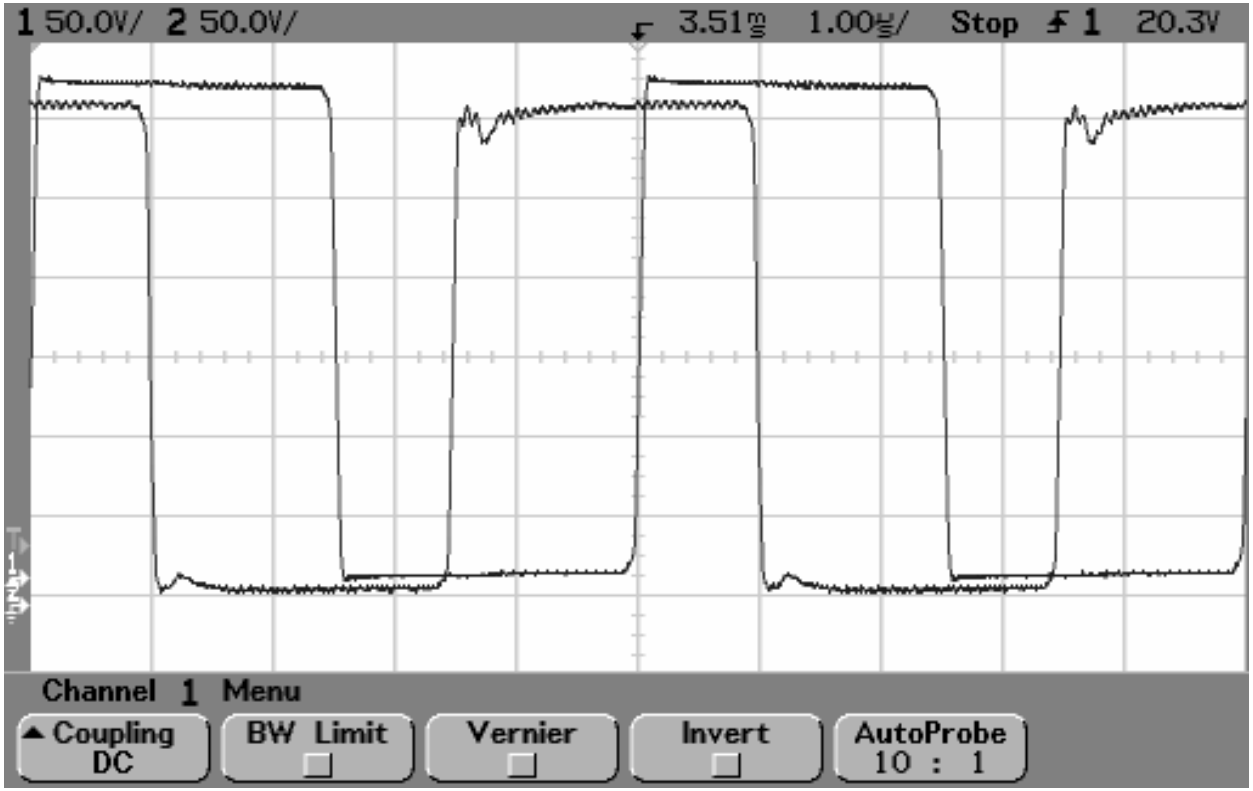
Experimental results

Interleaved Boost PFC



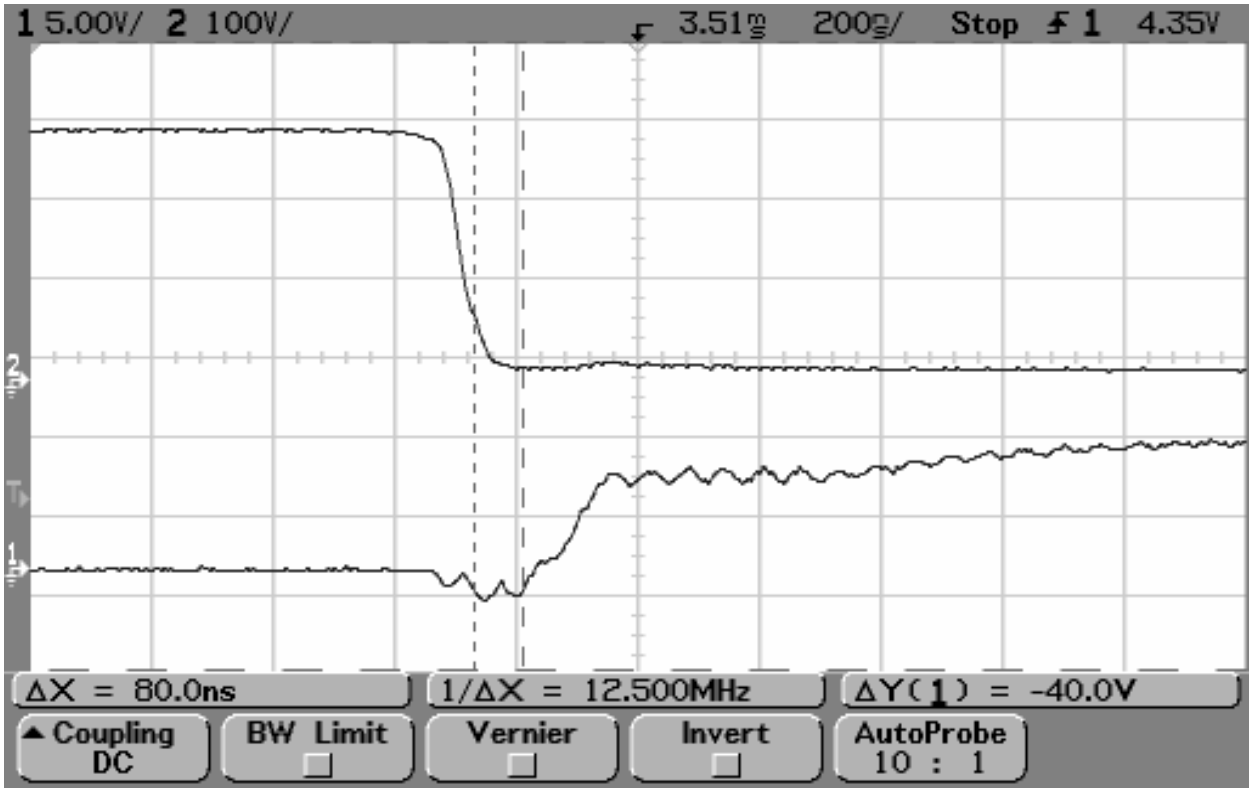
- 2 Interleaved boost converters
- MOSFET Rds,on current sense
- Excellent current sharing between modules

Phase Shifted Full Bridge



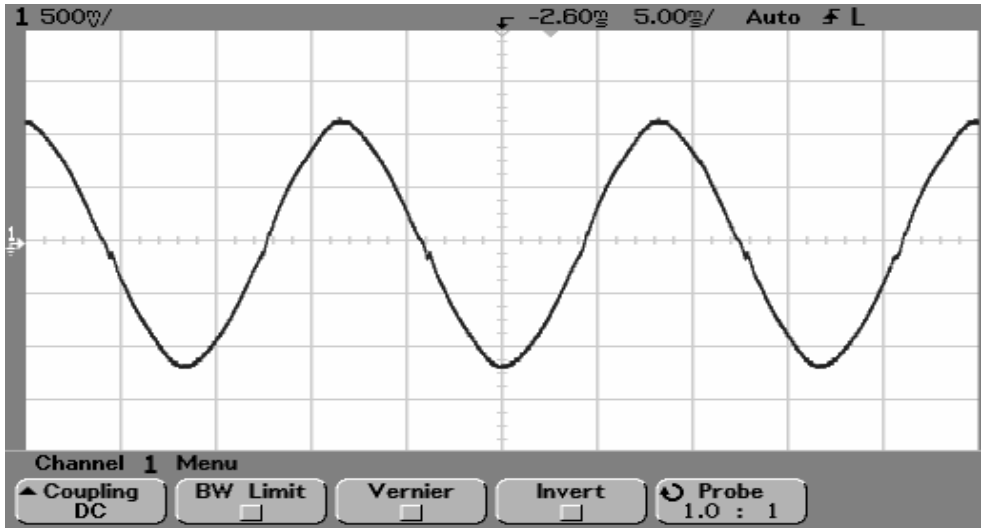
- DC-DC Phased-Shifted Full-Bridge operates off PFC boost
- Vds of Q6/Q7

Zero Voltage Switching

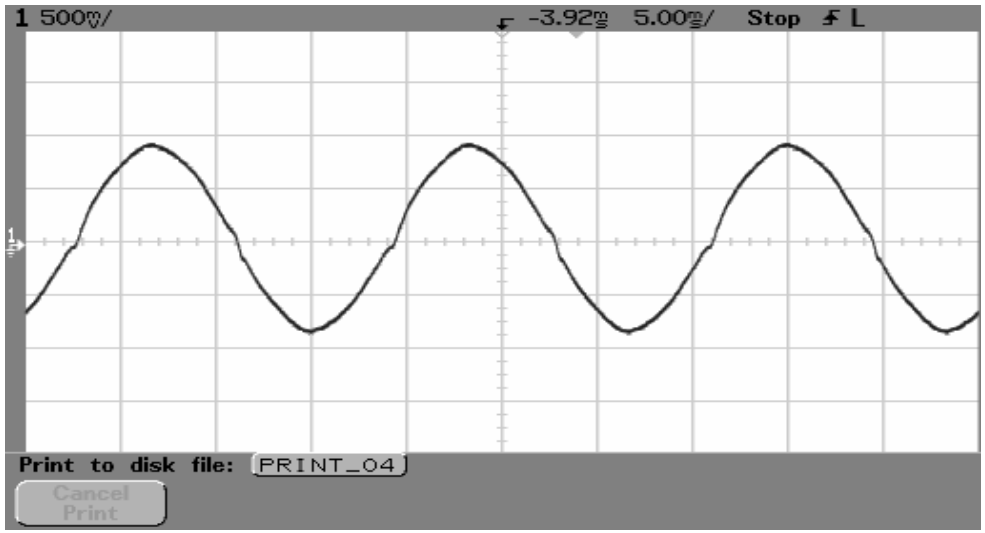


- ZVS of the PSFB
- Bottom trace is Vgs top trace is Vds.
- Vds falls to 0V before Vgs turns on MOSFET

DSP Controlled PFC Controller



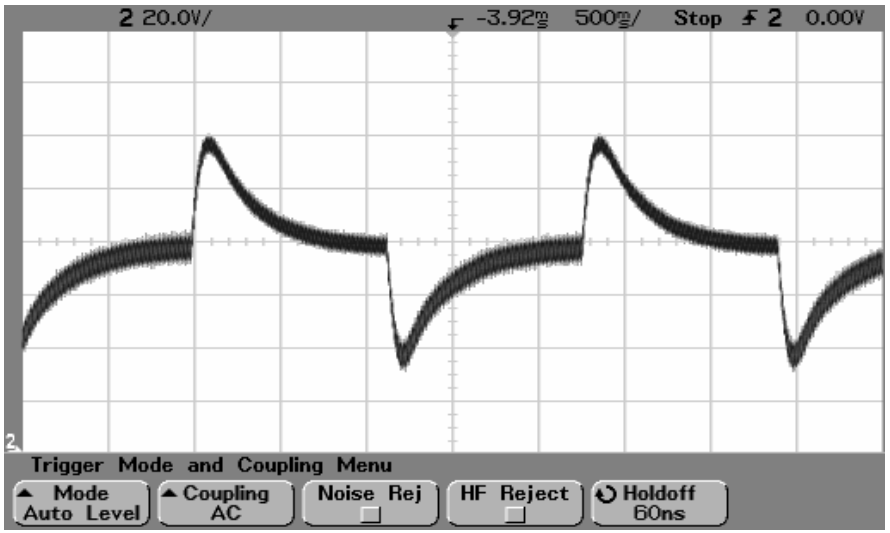
Input Current,
Po = 860W



Input Current,
Po = 580W

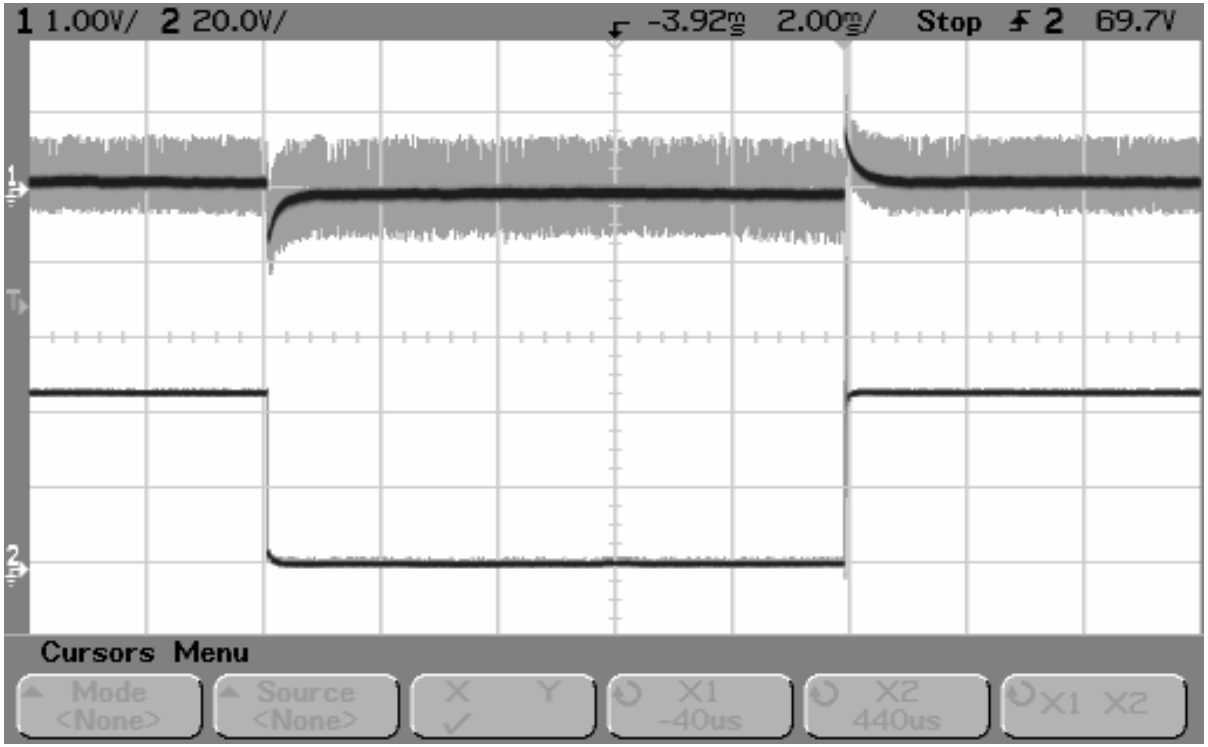
DSP Controlled PFC Controller

PFC MOSFETs Drain-Source Voltages



DC Bus Voltage Transient Response, Step load = 250W

DC/DC Stage Transient Response



- Load step = 230 W (580W → 350W → 580W)
- Voltage deviation = 1.6% @ 48V
- Settling time to within 1% = 250uS

Digital Rectifier Summary

- Digital control allows:
  - Sophisticated fault detection.
  - Supports wide voltage and load range (DCM and CCM).
  - Soft start and bring up sequence completely programmable.
  - Long timeframe load sharing and deadband control loops easily implemented.
  - Diagnostic data logging.
  - Single processor for communication and control.
- Performance very similar to best analog designs.

Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Next Steps

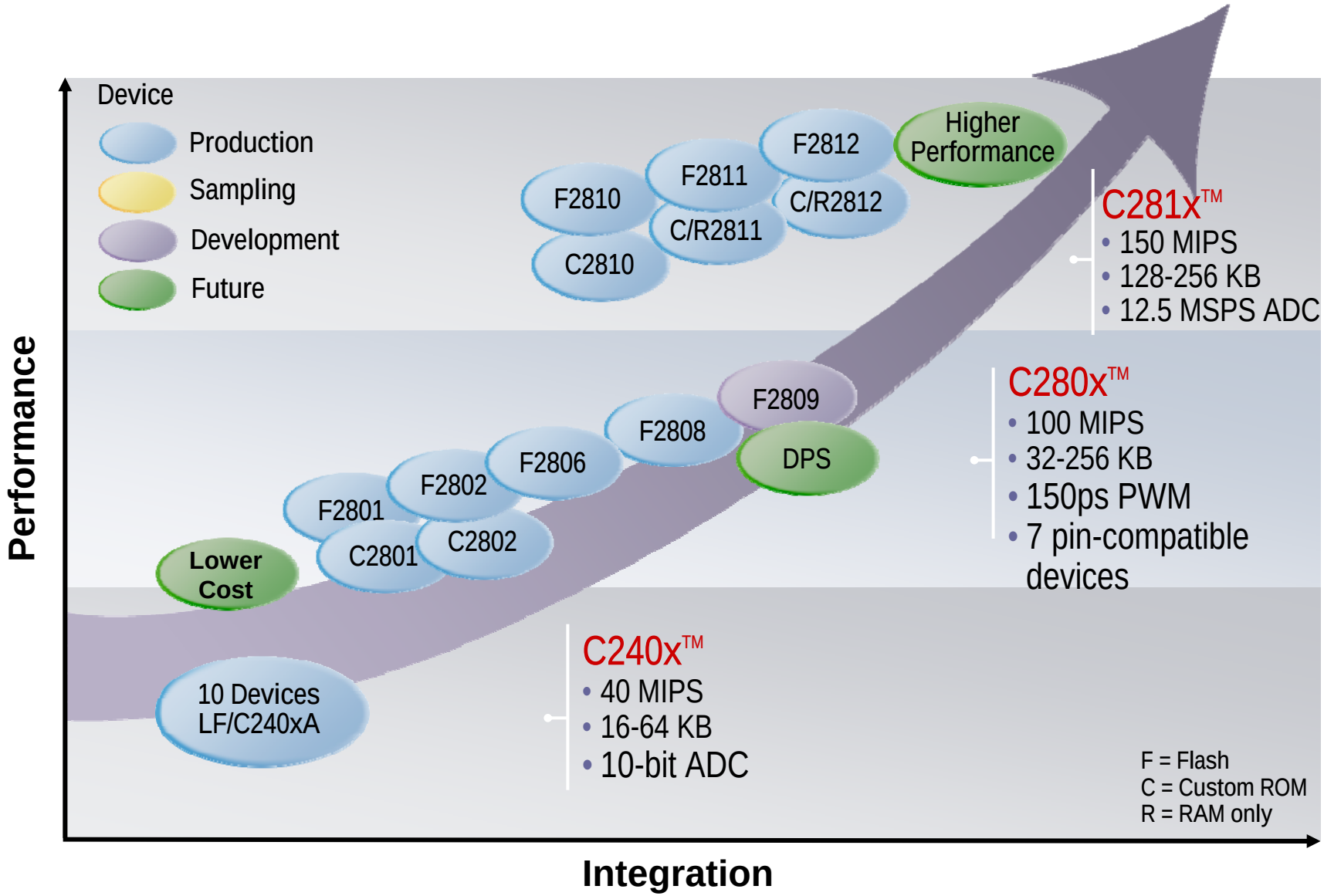
Real-Time debug

- RTDebug is a Non-intrusive debug scheme supported via on-chip H/W (utilizes spare / dead cycles in CPU buses)
- Allows user full interaction while application runs un-disturbed (at speed)
- Can interrogate / modify any memory, register, variable, ..etc
- Supports Single step / Break point in back-ground code while ISR (time critical loops + PWM) continues to run at speed.
- Clock / cycle profiling allows time critical code analysis.

Agenda

- The Digital Vision: Why DSP?
- Digital world: FACTS and FIGURES
- Digital AC/DC Rectifier challenges
- Software Strategy
- Implementation
- Next Steps

C2000™ Roadmap



How to get Started today?

Tools	▪ F2812 eZdsp™ & R2812 eZdsp developer's kit ($495) ▪ F2808 eZdsp kit in 1Q05 ($495)
Software	▪ Code Composer StudioTM IDE for C2000TM ($495 today) ▪ Application specific libraries ▪ Math functions ▪ Communications drivers ▪ Pre-bundled system solutions
Training and Support	▪ Control developers seminar ▪ DMC workshop ▪ One-day technical introduction to C28x™ ▪ Multi-day get started developing C28x™ workshop
High-Performance Analog	Numerous data converter and power management products designed for motor control
Third Party Network	▪ Development boards and emulation tools ▪ Large consultant network ▪ Increasing range of application software

TI Provides the Systems Expertise, Silicon, Software and Support for Control Applications

Current Voltage

Position/ Speed Feedback

Systems Expertise

Silicon

Power Electronics

TMS320C2000

Texas Instruments

EE Prom

TI Power Management

CAN Network

PC Test Equipment

Software

Digital Software Library

Kalman Filter

Precision PID

Filter Lib

Field Oriented Control

eXpressDSP™ Compliant DSP TEXAS INSTRUMENTS

Peripheral Drivers

"IQmath"

DSP/BIOS

Foundation Software

Support

Web casts

Workshops

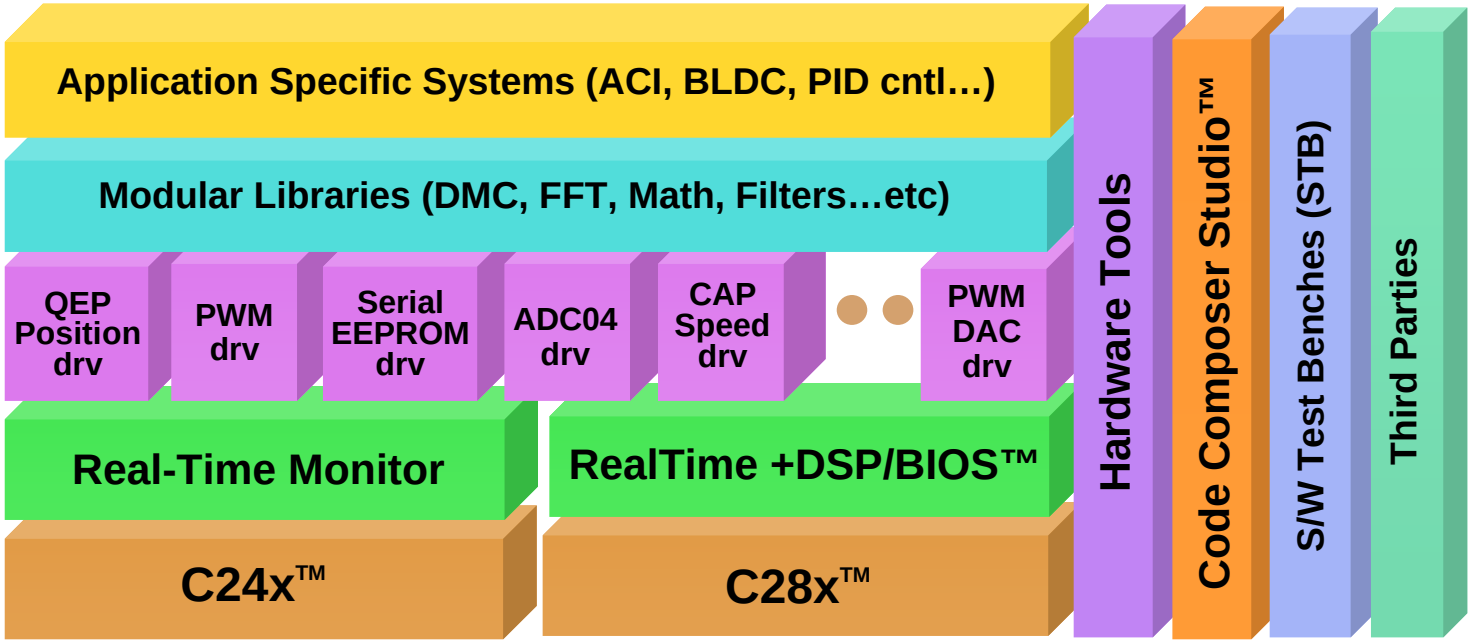
KnowledgeBase

Application notes

TI Third Party Network Member DSP TEXAS INSTRUMENTS

Modular Software Development for Digital Control Systems

All Modules Available in C/C++ Environment



- Reduces time to market
- Provides reuseable software

Gets you there quickly

http://www.ti.com/c2000appsw
http://www.ti.com/c2000sigproclib

The Foundation: Software Libraries

Motor Control Specific SW Modules

Forward and Inverse Clarke/Park Transforms,, BLDC Specific PWM Drivers, Leg Current Measurement Drivers, BLDC Commutation triggers, ACI Speed and Rotor Position Estimators, PID Controllers, Extended Precision PID Controllers.

Peripheral & Communication Drivers

SCI (UART) Packet Driver, Virtual SPI Drivers, Virtual I2C Drivers, Serial EEPROM Drivers, GPIO Driver.

Fixed Point Trigonometric and Log Routines

Fixed Point Sine, Cosine, Tangent routines, Square Root, Logarithm Functions. Reciprocal calculation.

IQ Math 32-Bit Virtual Floating Point Library

Multiply, Divide, Multiply with Rounding, Multiply with Rounding and Saturation, Square Root, Sine and Cosine, routines.

Signal Processing Functions

FIR (Generic order), FIR (10th order), FIR(20th order), FIR using circular buffers. 128, 256, and 512 point complex and real FFTs.

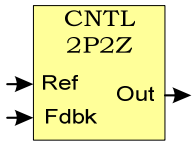
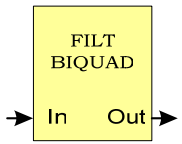
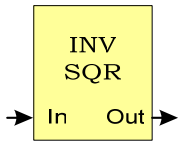
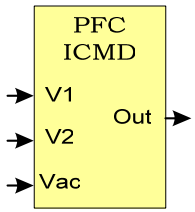
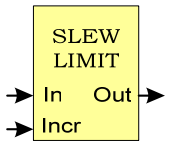
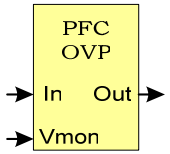
Signal Generator Functions

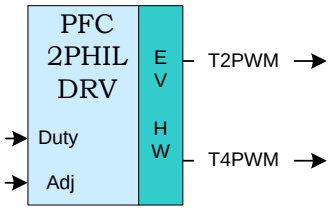
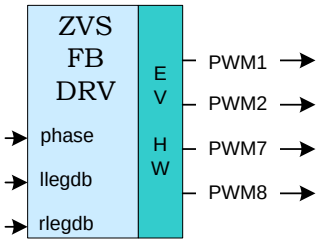
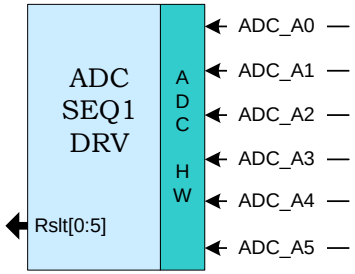
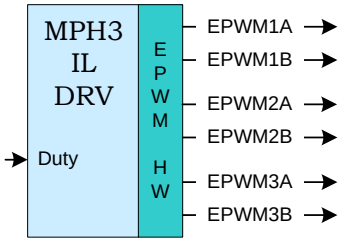
Sinewave generators, Ramp Generators, Trapezoidal Profile generators

Power Conversion Related Functions

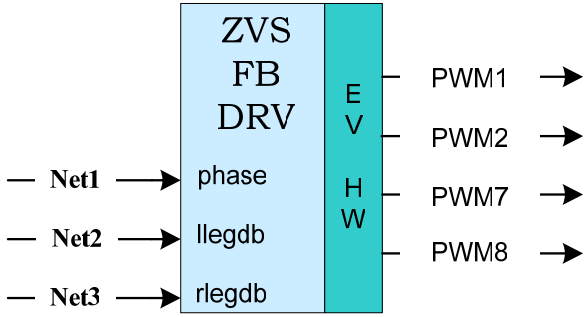
RMS computation, real power and apparent power computation, THD computation, PFC controllers.

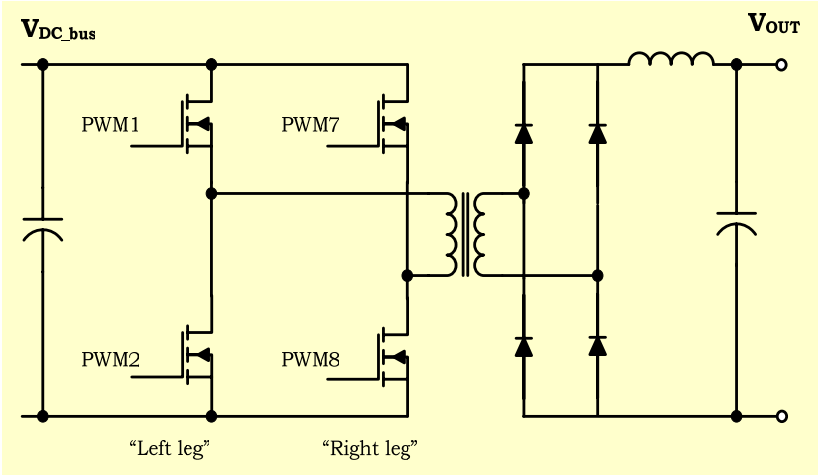
Digital Power Modules – Some Examples

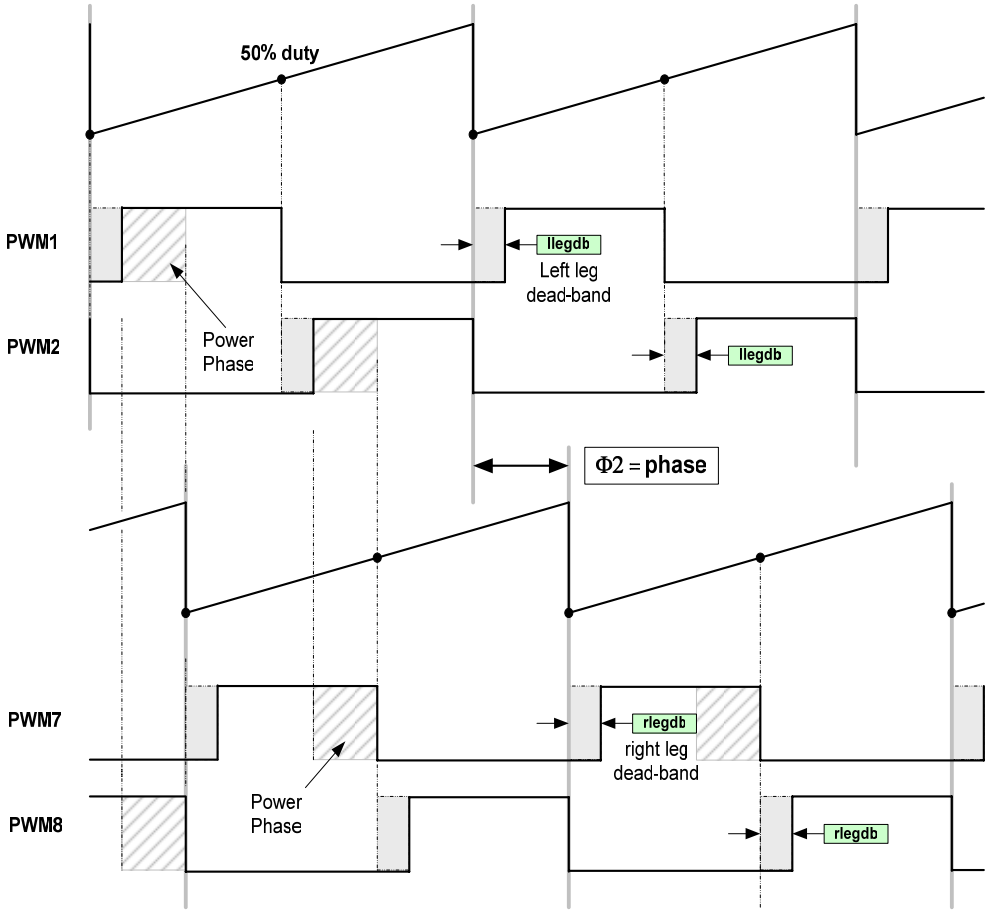
Symbol	Descr.	# Cycles
	Controller, 2 pole / 2 zero	36
	Biquad digital filter	46
	Inverse square function	78
	PFC Current Command function	30
	Slew rate Limiter function	17
	PFC over-voltage monitor	25

Symbol	Descr.	# Cycles
	PFC 2-phase Interleaved PWM s/w driver	26
	Zero Voltage Switched Full Bridge PWM s/w driver	14
	Analog / Digital conv. Sequencer s/w driver	57
	Multi-phase3 Interleaved PWM s/w driver	15

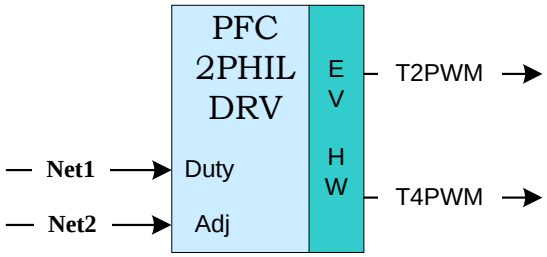
S/W driver module – ZVSFB

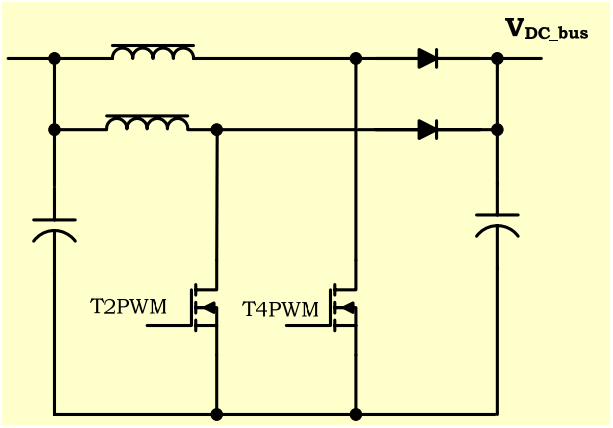


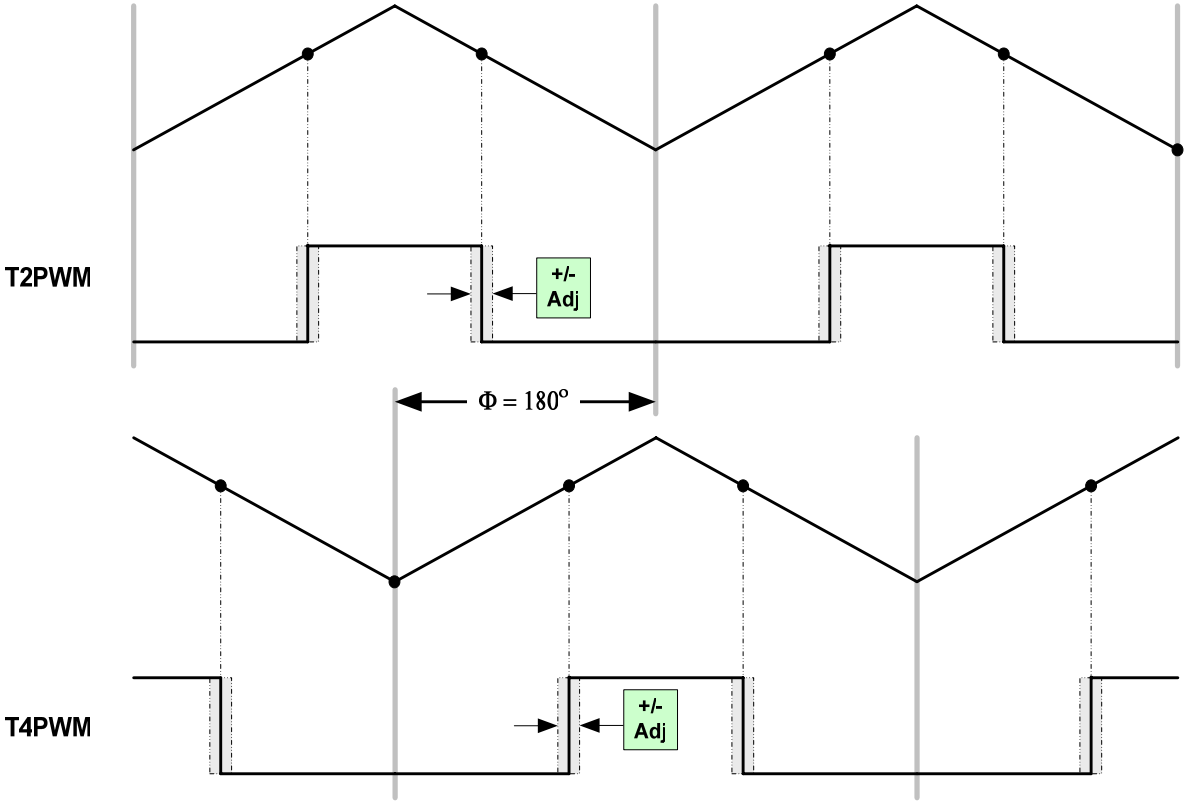




S/W driver module – PFC2PHIL







Digital Power Supplies: collaterals

- First version of the Digital Power Supply Library has been released

Digital Power System Solutions for C2808, C2806, C2802 and C2801		
System	**Description**	**Part Number**
DC-DC Buck Converter	DC-DC Buck Converter using High Resolution ePWM	SPRC229
High Resolution ePWM	Demonstrates HRPWM Capabilities for Digital Power Applications	SPRC227
Standard ePWM	Demonstrates ePWM Capabilities for Digital Power Applications	SPRC228

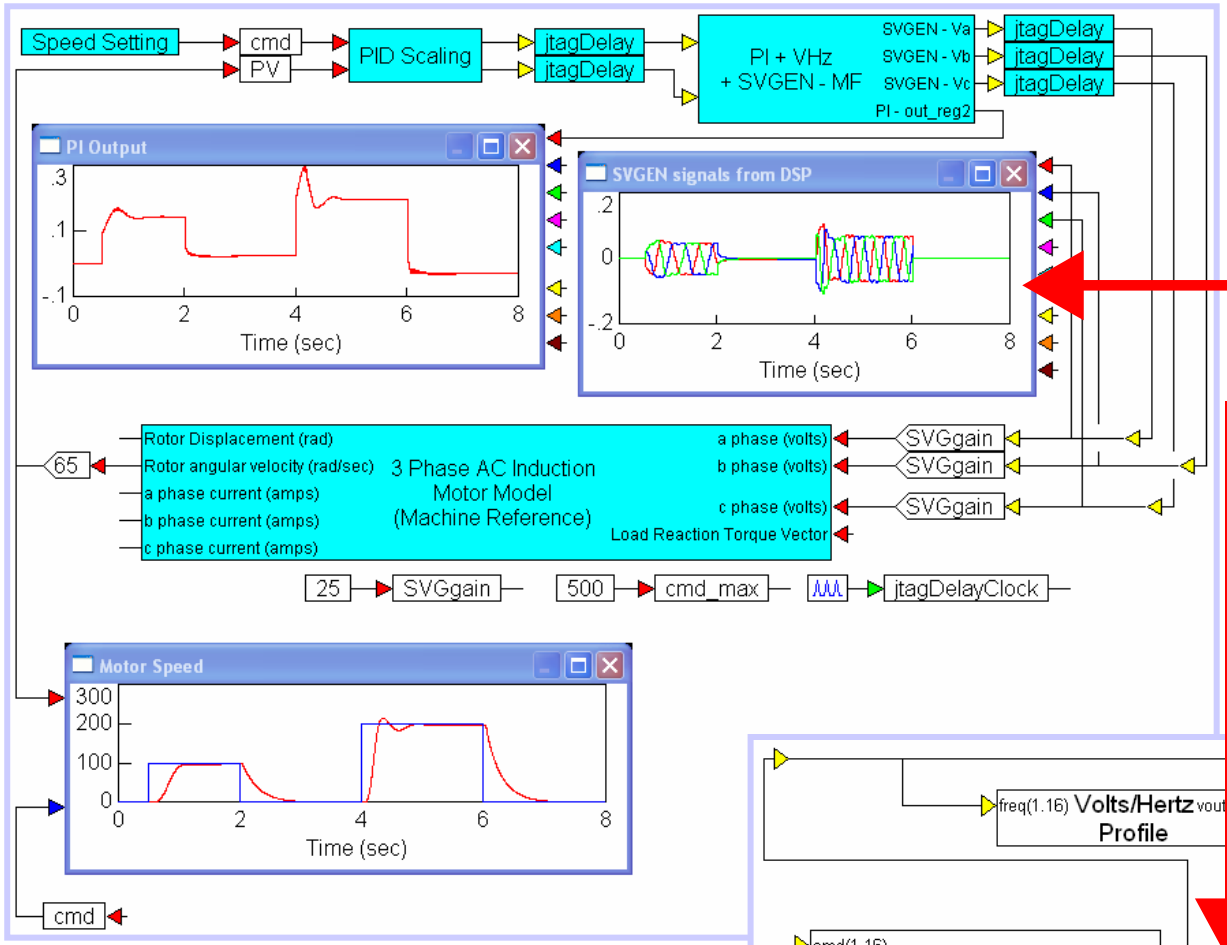
- Check out: www.ti.com/c2000appsw

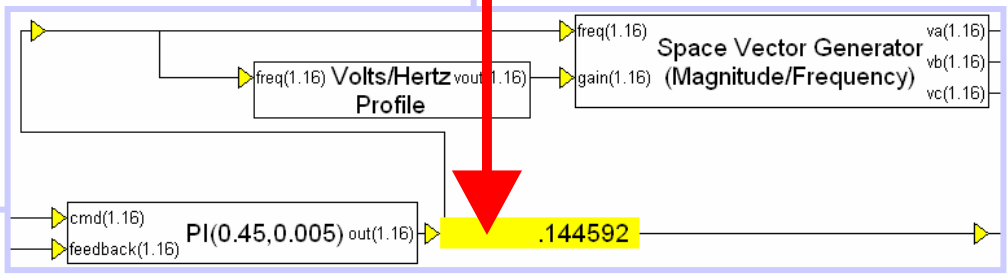
- Digital Power Theory application note: **SPRAAB3**

Graphical Ease of Development

Visual Solutions INCORPORATED

Modeling the Future





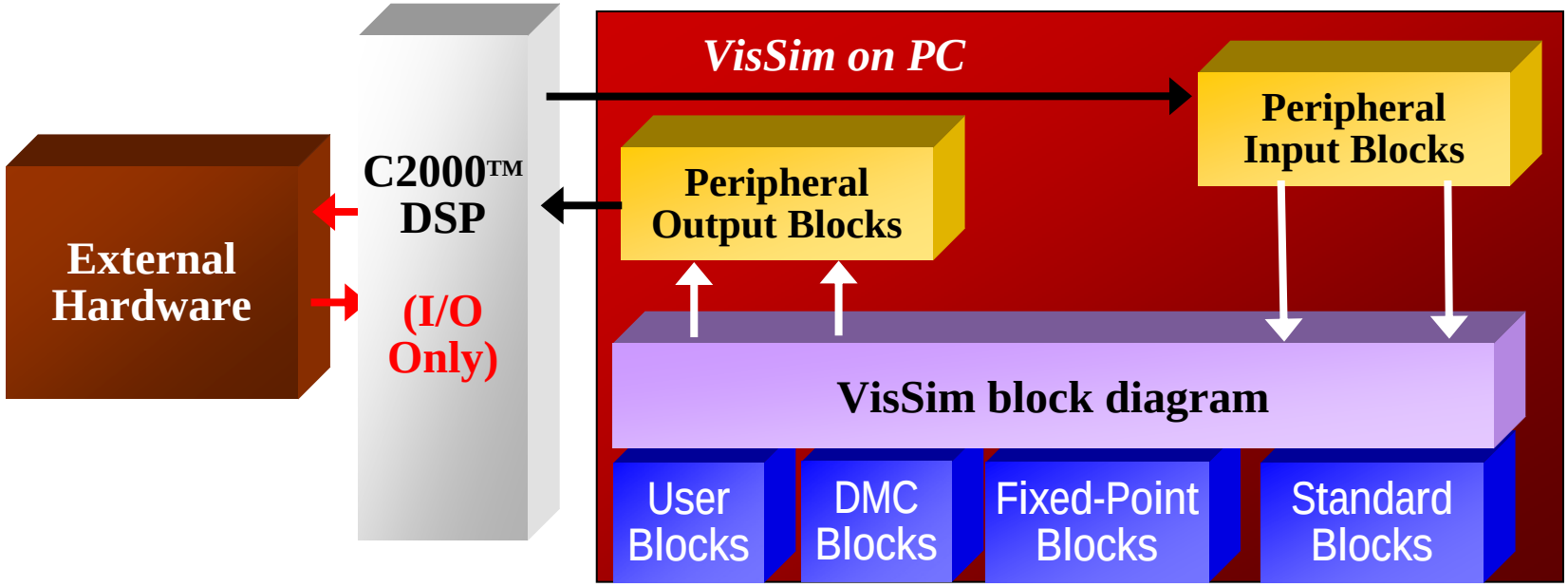
- Use high-level, pre-debugged blocks
- Support simulation of controller at block level on PC
- Allow mouse probe of every input and output to display values at any instant
- Debug block-level simulation on PC

Hardware-in-the-Loop

Visual Solutions INCORPORATED

Modeling the Future

- Pure simulation plus DSP-in-loop simulation and block level monitoring gives rapid feedback of controller response

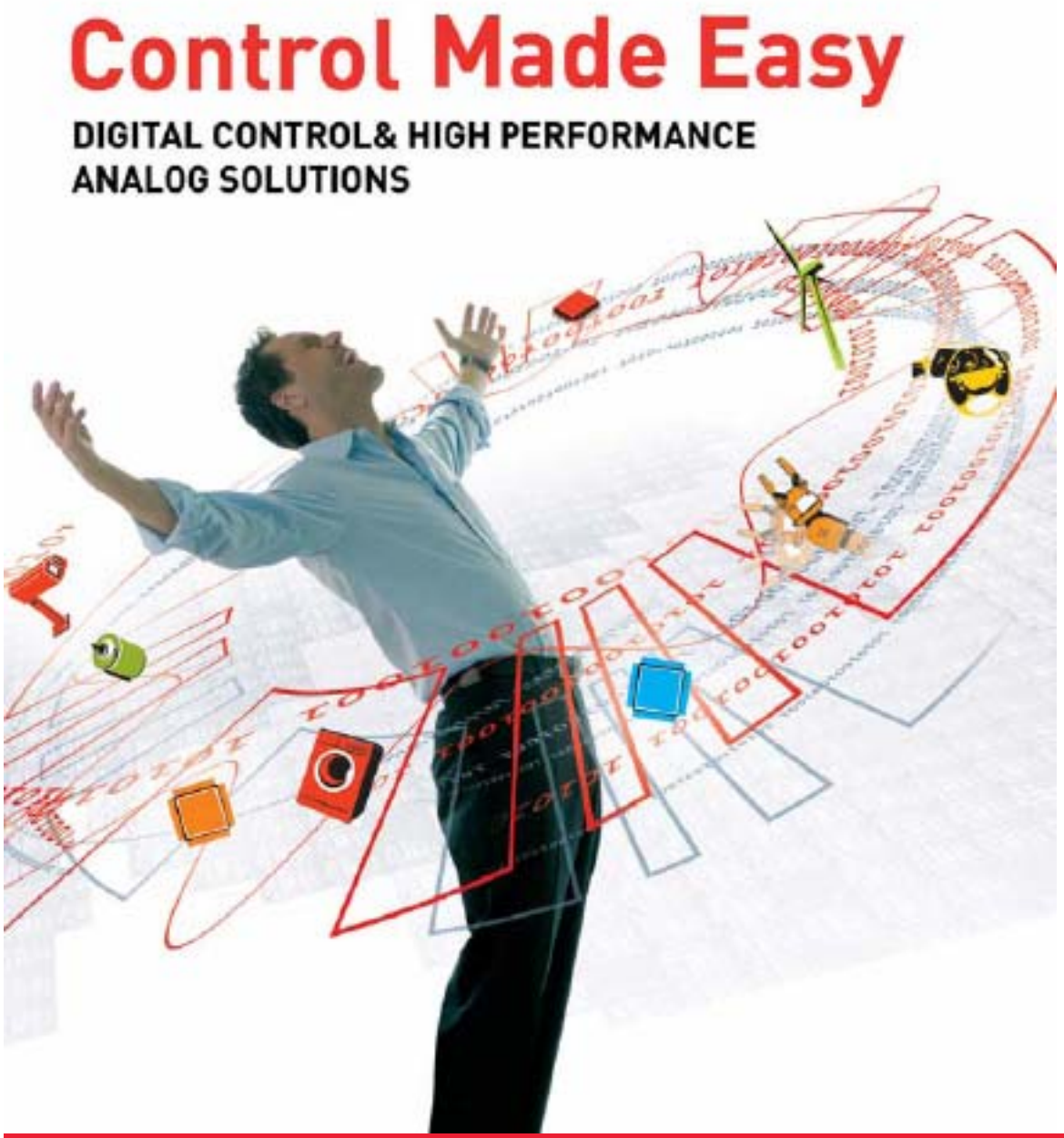


Test DSP based controller against virtual plant on PC using JTAG HotLink

- Inject plant failure modes to test controller response
- High/Low watermark on fixed-point blocks gives numerical “headroom” safety factor
- Interactive DSP utilization gives continuous CPU load factor
- Interactively Change DSP controller gains from VisSim and plot DSP response.

Conclusion

- Digital Power is the future
- One of the Traditional Fear Factors industry is SW complexity
- TI is enabling Digital Power with the Digital Power Supply Library
- 28x has the right set of peripherals for Digital Power Supply
- Tools, Documentation are in place to start today
- TI can support for a complete design with Analog and Digital products



Thank you

Visit us
Stand 134, Hall 12

Meet our experts and discover our expanded range of products

Third parties exhibiting at TI booth:

Active DSP
Visual Solutions
INCORPORATED
Modeling the Future

TECHNOSOFT