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

Diagram showing the components of an AC/DC rectifier system, including:
- AC input
- Filter Bridge
- PFC (Power Factor Correction) Control
- DC/DC Control
- Output

Additional components include:
- Inrush/Hot-plug Control
- Interface circuit
- Monitor (MCU ?)
- Aux P/S
- PFC Control
- Current/Load Sharing Control
- DC/DC Converter Control
- Multi-mode Power control
- Supervisory House Keeping Circuits
- UART
- To Host

Technology for Innovators™

Texas Instruments
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

- CPU (DSP/uC/RISC) + Memory (FLASH/ROM, RAM)
- ADC
- PWM (‘DAC’ function)
- Comms: CAN, UART, SPI, IIC, IIS, FlexRay, USB, EMAC
- Control Loop (i.e. PID/IIR)
- Peripherals Shown, Found On TI DSP C2000 Family Of Devices
- RX | 0101010101010101 |
- TX | 01010101010101 |
- i.e. Buck Converter
- i.e. V
- i.e. Encoder
- i.e. Hall Sensor
### Why DSP for Power Supplies?

<table>
<thead>
<tr>
<th>Controller</th>
<th>PWM</th>
<th>Power Elec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog or Digital ??</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Analog Controller

- High bandwidth
- High resolution
- Easy to understand / use
- "relatively’ low cost ??

#### Digital Controller

- 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 | Bandwidth limitations (sampling loop) |
| Component tolerances | PWM frequency and resolution limits |
| Hardwired / not flexible | Numerical problems (quantisation, rounding,...) |
| Limited to classical control theory only | AD / DA boundary (resolution, speed, cost) |
| Large parts count for complex systems | CPU performance limitations |
| System cost |                             |

---

**Why DSP for Power Supplies?**

Technology for Innovators™

[Texas Instruments]
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……

280x – DSP

• Clock speed (MIPs)
• Word size (dynamic range)
• MAC size (16x16 / 32x32)
• Large on-chip SRAMs
• C / C++

"DAC"

ADC

PWM

Continually Improving specs!

Outputs
• Buck / Boost
• Half bridge
• Full bridge / PS
• Multi phase IL

Inputs
• Current
• Voltage
• Temperature

• Resolution
• Linearity / Accuracy
• Sampling rate (speed)
• HV isolation

"DAC"

Technology for Innovators®
# Processor capability

## PWM freq. vs PWM per.

<table>
<thead>
<tr>
<th>PWM freq. (KHz)</th>
<th>PWM per. (uS)</th>
<th>Processor MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>20.0</td>
<td>800 2000 3000</td>
</tr>
<tr>
<td>100</td>
<td>10.0</td>
<td>400 1000 1500</td>
</tr>
<tr>
<td>200</td>
<td>5.0</td>
<td>200 500 750</td>
</tr>
<tr>
<td>250</td>
<td>4.0</td>
<td>160 400 600</td>
</tr>
<tr>
<td>300</td>
<td>3.3</td>
<td>133 333 500</td>
</tr>
<tr>
<td>500</td>
<td>2.0</td>
<td>80 200 300</td>
</tr>
<tr>
<td>750</td>
<td>1.3</td>
<td>53 133 200</td>
</tr>
<tr>
<td>1000</td>
<td>1.0</td>
<td>40 100 150</td>
</tr>
</tbody>
</table>

**MIPS = Million Instruction Per Second**

---

## # Inst. vs Algorithm

<table>
<thead>
<tr>
<th>S/W algorithm</th>
<th>clks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller (2 pole / 2 zero)</td>
<td>26</td>
</tr>
<tr>
<td>Controller (3 pole / 3 zero)</td>
<td>36</td>
</tr>
<tr>
<td>PFC current command</td>
<td>30</td>
</tr>
<tr>
<td>PFC OVP</td>
<td>25</td>
</tr>
<tr>
<td>BiQuad Filter</td>
<td>46</td>
</tr>
<tr>
<td>ZVSFB PWM driver</td>
<td>14</td>
</tr>
<tr>
<td>PFC2PHIL PWM driver</td>
<td>26</td>
</tr>
</tbody>
</table>

# Technology for Innovators™

[Logo: Texas Instruments]
## Typical Power Stage Switching Frequencies

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

### 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

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

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

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

\[ f(x) = \sin(x) \]

Out = \sin(In)
Exploring Modularity

**Module example 2**
- Single In / Single out
- Configurable
- \( m, b, \) Constant ? or Variable ?
- No History
- Multiple Instantiation

\[
f(x) = mx + b
\]

\[
\text{Line}
\]

\[
\text{Out} = m \cdot \text{In} + b
\]

**Module example 3**
- Single In / Single out
- Non-Configurable
- History
- Multiple Instantiation

\[
f(x) = \left( x_n + x_{n-1} + x_{n-2} + x_{n-3} \right) / 4
\]

\[
\text{BoxCarAvg}
\]

\[
\text{Out} = \frac{X(n) + X(n-1) + X(n-2) + X(n-3)}{4}
\]
Module Types

**Application Indep. / Peripheral Indep.**
- INV
- SQR
- In
- Out

**Application Config. / Peripheral Indep.**
- CNTL
- 2P2Z
- Ref
- Out
- Fdbk

**Application Config. / Peripheral Depend. (“Peripheral Driver”)**
- ZVS
- FB
- DRV
- PWM1
- PWM2
- PWM7
- PWM8

- phase
- ilegdb
- rlegdb

- MPH3
- IL
- DRV
- EPWM1A
- EPWM1B
- EPWM2A
- EPWM2B
- EPWM3A
- EPWM3B
Peripheral Drivers

- **Vref (Q15)**
  - Ref
  - Out

- **Duty (Q15)**
  - CNTL 2P2Z
    - Ref
    - Fdbk
  - FILT 2P2Z
    - Out
    - In

- **Vout (Q15)**

- **BUCK DRV**
  - EPWM
  - HW
  - EPWM1A

- **ADC SEQ1 DRV**
  - ADC
    - HW
    - Rs1t0
    - Rs1t1
    - Rs1t2
    - Rs1t3
  - ADC_A0
  - ADC_A1
  - ADC_A2
  - ADC_A3

**CPU dependency only:**
- Math / algorithms
- Per-Unit math (0-100%)
- Independent of Hardware

**Dependencies:**
- PWM frequency
- System clock frequency
- # ADC bits (10 / 12 ?)
- Unipolar, Bipolar ?
- Offset ?
## Q-Math Representation

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

<table>
<thead>
<tr>
<th>Qn</th>
<th>Format</th>
<th>Value Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q15</td>
<td>S.FFF FFFF FFFF FFFF FFFF</td>
<td>- 1 &lt; N &lt; +0.99999...</td>
</tr>
<tr>
<td>Q14</td>
<td>SI.FF FFFF FFFF FFFF FFFF</td>
<td>- 2 &lt; N &lt; +1.99999...</td>
</tr>
<tr>
<td>Q13</td>
<td>SII.F FFFF FFFF FFFF FFFF</td>
<td>- 4 &lt; N &lt; +3.99999...</td>
</tr>
<tr>
<td>Q12</td>
<td>SIII.F FFFF FFFF FFFF FFFF</td>
<td>- 8 &lt; N &lt; +7.99999...</td>
</tr>
<tr>
<td>Q0</td>
<td>SIII IIII IIII IIII IIII</td>
<td>-32,768 &lt; N &lt;+32,767</td>
</tr>
</tbody>
</table>

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

SSI.F FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF (32 bit format)
SI.FF FFFF FFFF FFFF FFFF (adjusted for Q14, 16 bit format)

e.g. Q15 x Q15 = Q30

SS.FF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF (32 bit format)
S.FFF FFFF FFFF 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!
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

How complex?
How much code development?
How much maintenance burden?
How wasteful on memory?

3. ASM Macros – great for modularity!

Modern compilers support:
- Macro parameter passing
- Macro variable & label 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

Number of Instructions / cycles (words)

<table>
<thead>
<tr>
<th>PWM (KHz)</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>250</td>
<td>400</td>
</tr>
<tr>
<td>300</td>
<td>333</td>
</tr>
<tr>
<td>350</td>
<td>286</td>
</tr>
<tr>
<td>400</td>
<td>250</td>
</tr>
<tr>
<td>500</td>
<td>200</td>
</tr>
</tbody>
</table>

% impact per 10 instructions

<table>
<thead>
<tr>
<th>PWM (KHz)</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>200</td>
<td>2.0%</td>
</tr>
<tr>
<td>250</td>
<td>2.5%</td>
</tr>
<tr>
<td>300</td>
<td>3.0%</td>
</tr>
<tr>
<td>350</td>
<td>3.5%</td>
</tr>
<tr>
<td>400</td>
<td>4.0%</td>
</tr>
<tr>
<td>500</td>
<td>5.0%</td>
</tr>
</tbody>
</table>

Fear factor!
Exploring Ideas / Methods

4. Pointer based parameter passing (data flow)

Conventional approach

\[ \text{Out3} = f3(\text{In3A}, \text{In3B}) \]

<table>
<thead>
<tr>
<th>Function</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f1 )</td>
<td>( \text{In1A}, \text{In1B} )</td>
<td>( \text{Out1} )</td>
</tr>
<tr>
<td>( f2 )</td>
<td>( \text{In2A} )</td>
<td>( \text{Out2} )</td>
</tr>
<tr>
<td>( f3 )</td>
<td>( \text{In3A}, \text{In3B} )</td>
<td>( \text{Out3} )</td>
</tr>
</tbody>
</table>

Pointer based approach

\[ \text{Pseudo code} \]

\[
\begin{align*}
\text{move } \_& , \text{In1A} \quad (2) \\
\text{move } \_& , \text{In1B} \quad (2) \\
\text{call } f1 & \quad (8) \\
\text{move } \_& , \text{In2A} \quad (2) \\
\text{call } f2 & \quad (8) \\
\text{move } \text{Out1}, \text{In3A} & \quad (2) \\
\text{move } \text{Out2}, \text{In3B} & \quad (2) \\
\text{call } f3 & \quad (8) \\
\text{move } \text{Out3}, \_ & \quad (2)
\end{align*}
\]

\[ \text{Pseudo code (without macros)} \]

\[
\begin{align*}
\text{call } f1 & \quad (8) \\
\text{call } f2 & \quad (8) \\
\text{call } f3 & \quad (8)
\end{align*}
\]

\[ \text{Pseudo code (with macros)} \]

\[
\begin{align*}
\text{call } f1 & \quad (\text{zero}) \\
\text{call } f2 & \quad (\text{zero}) \\
\text{call } f3 & \quad (\text{zero})
\end{align*}
\]
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**

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

| 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%    |

<table>
<thead>
<tr>
<th>BG</th>
<th>Function / Activity</th>
<th># inst.</th>
<th>Tot.Cyc.</th>
<th>Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Comms + Supervisory</td>
<td>400</td>
<td>434</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ Soft-Start + Other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SLEW_LIMIT 1</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SLEW_LIMIT 2</td>
<td>17</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

% ISR utilization = 87%
Spare ISR MIPS = 42.6
BG loop rate (KHz) / (uS) = 29.0 34.4
Experimental results
Interleaved Boost PFC

- 2 Interleaved boost converters
- MOSFET $R_{ds,on}$ 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% = 250μS
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

Performance

Device
- Production
- Sampling
- Development
- Future

Higher Performance
- F2812
- C/R2812

C281x™
- 150 MIPS
- 128-256 KB
- 12.5 MSPS ADC

C280x™
- 100 MIPS
- 32-256 KB
- 150ps PWM
- 7 pin-compatible devices

C240x™
- 40 MIPS
- 16-64 KB
- 10-bit ADC

Lower Cost
- F2801
- C2801

10 Devices
- LF/C240xA

Technology for Innovators™

Texas Instruments
### How to get Started today?

<table>
<thead>
<tr>
<th>Tools</th>
</tr>
</thead>
</table>
|   • F2812 eZdsp™ & R2812 eZdsp developer’s kit ($495)  
|   • F2808 eZdsp kit in 1Q05 ($495)  |

<table>
<thead>
<tr>
<th>Software</th>
</tr>
</thead>
</table>
|   • Code Composer StudioTM IDE for C2000TM ($495 today)  
|   • Application specific libraries  
|   • Math functions  
|   • Communications drivers  
|   • Pre-bundled system solutions  |

<table>
<thead>
<tr>
<th>Training and Support</th>
</tr>
</thead>
</table>
|   • Control developers seminar  
|   • DMC workshop  
|   • One-day technical introduction to C28x™  
|   • Multi-day get started developing C28x™ workshop  |

<table>
<thead>
<tr>
<th>High-Performance Analog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerous data converter and power management products designed for motor control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Third Party Network</th>
</tr>
</thead>
</table>
|   • 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

Systems Expertise
Silicon
Software

Support
Web casts
Workshops
KnowledgeBase
Application notes

Current Voltage
Position/Speed Feedback

Power Electronics
TI Power Management
EE Prom

Kalman Filter
Precision PID
Filter Lib
Field Oriented Control

“IQmath”
Peripheral Drivers
DSP/BIOS
Foundation Software

CAN Network
PC Test Equipment

TMS320C2000
Texas Instruments

Technology for Innovators®
Modular Software Development for Digital Control Systems

All Modules Available in C/C++ Environment

- **Application Specific Systems (ACI, BLDC, PID cntl...)**
- **Modular Libraries (DMC, FFT, Math, Filters...etc)**
- **QEP Position drv**
- **PWM drv**
- **Serial EEPROM drv**
- **ADC04 drv**
- **CAP Speed drv**
- **PWM DAC drv**

**Real-Time Monitor**

**RealTime +DSP/BIOS™**

**C24x™**

**C28x™**

**Hardware Tools**

**Code Composer Studio™**

**S/W Test Benches (STB)**

**Third Parties**

- Reduces time to market
- Provides reusable software

Gets you there quickly

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

Technology for Innovators™
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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Descr.</th>
<th># Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTL 2P2Z</td>
<td>Controller, 2 pole / 2 zero</td>
<td>36</td>
</tr>
<tr>
<td>FILT BIQUAD</td>
<td>Biquad digital filter</td>
<td>46</td>
</tr>
<tr>
<td>INV SQR</td>
<td>Inverse square function</td>
<td>78</td>
</tr>
<tr>
<td>PFC ICMD</td>
<td>PFC Current Command function</td>
<td>30</td>
</tr>
<tr>
<td>SLEW LIMIT</td>
<td>Slew rate Limiter function</td>
<td>17</td>
</tr>
<tr>
<td>PFC OVP</td>
<td>PFC over-voltage monitor</td>
<td>25</td>
</tr>
</tbody>
</table>

### Symbol

- **PFC 2-phase Interleaved PWM s/w driver**: 26 cycles
- **Zero Voltage Switched Full Bridge PWM s/w driver**: 14 cycles
- **Analog / Digital conv. Sequencer s/w driver**: 57 cycles
- **Multi-phase3 Interleaved PWM s/w driver**: 15 cycles

---

**Technology for Innovators™**
S/W driver module – ZVSFB

- Net1: phase
- Net2: I_legdb
- Net3: R_legdb

\[ V_{DC\_bus} \rightarrow V_{OUT} \]

\[ PWM1 \rightarrow \]
\[ PWM2 \rightarrow \]
\[ PWM7 \rightarrow \]
\[ PWM8 \rightarrow \]
S/W driver module – PFC2PHIL

![Diagram of S/W driver module – PFC2PHIL](image)
Digital Power Supplies: collaterals

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

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-DC Buck Converter</td>
<td>DC-DC Buck Converter using High Resolution ePWM</td>
<td>SPRC229</td>
</tr>
<tr>
<td>High Resolution ePWM</td>
<td>Demonstrates HRPWM Capabilities for Digital Power Applications</td>
<td>SPRCZ77</td>
</tr>
<tr>
<td>Standard ePWM</td>
<td>Demonstrates ePWM Capabilities for Digital Power Applications</td>
<td>SPRC228</td>
</tr>
</tbody>
</table>

◆ Check out: [www.ti.com/c2000appsw](http://www.ti.com/c2000appsw)

◆ Digital Power Theory application note: **SPRAAB3**
Graphical Ease of Development

- 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

- 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

Modeling the Future

Technology for Innovators