© 2017 Yizhe Zhang
A SERIES-STACKED ARCHITECTURE WITH ISOLATED
SERVER-TO-BUS CONVERTERS FOR
HIGH-EFFICIENCY DATA CENTER POWER DELIVERY

BY

YIZHE ZHANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Assistant Professor Robert Pilawa-Podgurski
Abstract

Series-stacked server power delivery architectures have been proposed recently that can achieve much higher energy efficiency than conventional power delivery architectures. When servers are connected in series, differential power processing (DPP) converters can be used to regulate the server voltages when the servers are consuming different amounts of current. Server-to-bus DPP architecture has unique advantages among several other DPP architectures such as being able to achieve the minimum power processed in the DPP converters, and having a higher reliability than other DPP architectures. This work presents the development of a server-to-bus DPP architecture for server power delivery. The hardware prototype is built with four 4-to-1 isolated DPP converters with GaN switches. Four 12V 120W Dell servers are used in the bench test to validate the operation of server-to-bus DPP. 98.99% efficiency is achieved while the servers are running a real-life data center computational load.
To my parents, for their love and support.
I would like to thank my parents for all your love and support. Your encouragement and wisdom make me a better person and help me get through the most difficult times in my life. I want to thank my advisor Professor Robert Pilawa. You not only guide me to be a good electrical engineer and researcher, but also influence me to become an optimistic and kind man. Many thanks to all the members of the Pilawa group. You make our office and lab like a warm home where I wish to stay day and night. I am especially grateful to Enver Candan, who provided me precious and the most direct mentoring in my research. I cannot have made such fast progress in my research project without your strong support. I would also like to thank Professor Grace Gao, with whom I worked for one semester. Your trust in me and your kindness really helped me a lot. Moreover, I want to thank all my friends at UIUC. Your companionship makes my life here more colorful and meaningful. Finally I would like to thank the Power Group and ECE Illinois. I could not have become the person I am now without such a supportive environment.
# Table of Contents

List of Tables .......................................................... vi
List of Figures ............................................................ vii

Chapter 1 INTRODUCTION .............................................. 1

Chapter 2 SERVER-TO-BUS DPP ARCHITECTURE ................. 4
  2.1 Minimum Total Processed Power ............................ 4
  2.2 High Reliability ................................................. 8

Chapter 3 CONTROL FOR SERVER-TO-BUS DPP ARCHITECTURE .... 9
  3.1 Voltage Hysteresis Control .................................. 9
  3.2 Optimal String Current Control ......................... 9

Chapter 4 PROTOTYPE DPP HARDWARE FOR THE SERVER-TO-BUS ARCHITECTURE .......... 11
  4.1 Key Features of the GaN-Based DPP Converter Operation .. 11
  4.2 Power Loss Analysis .......................................... 20

Chapter 5 EXPERIMENTAL VERIFICATION .......................... 26
  5.1 Testbed ......................................................... 26
  5.2 Experimental Results ....................................... 28

Chapter 6 CONCLUSION .................................................. 34

References .................................................................... 35

Appendix A PROTOTYPE DPP HARDWARE COMPONENT LIST AND PCB LAYOUT ............ 39

Appendix B KEY MICROCONTROLLER CODE .......................... 50
List of Tables

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Server currents in the example where n is even</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Server currents in the example where n is odd</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Resistances of the inductor, transformer windings, and the switches</td>
<td>20</td>
</tr>
<tr>
<td>4.2</td>
<td>Conduction loss of the inductor, transformer, and switches</td>
<td>20</td>
</tr>
<tr>
<td>4.3</td>
<td>Loss breakdown of the DAB converter</td>
<td>25</td>
</tr>
<tr>
<td>5.1</td>
<td>Breakdown of the average input and output powers, the average power loss,</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>the system-level efficiency and power conversion efficiency during the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>experiment</td>
<td></td>
</tr>
<tr>
<td>A.1</td>
<td>The key components of the DPP converter</td>
<td>39</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Three DPP architectures. .............................................. 3
2.1 The $f(i_{\text{string}})$, $m$ and $(2m - n)$ in the example where $n$ is even. 7
2.2 The $f(i_{\text{string}})$, $m$ and $(2m - n)$ in the example where $n$ is odd. 8
3.1 Bi-directional hysteresis shape [1]. ................................. 10
4.1 The simplified schematic of the prototype DPP hardware. . . . . . 12
4.2 The prototype DPP hardware. ........................................ 12
4.3 The efficiency plot of the GaN-based DAB converter. ............ 13
4.4 The schematics of the DAB convertor. .............................. 14
4.5 Voltage and current waveforms of the DAB convertor under ideal conditions. ......................................................... 15
4.6 Detailed schematic of the first leg of the primary side full bridge. ................................................................. 15
4.7 Detailed schematic of one leg of the secondary side full bridge. 17
4.8 Oscilloscope screenshot of the DAB convertor operation in a real experiment. ......................................................... 18
4.9 Zoomed-in oscilloscope screenshot of the DAB convertor operation. ................................................................. 19
5.1 The testbed for the server-to-bus architecture. .................... 27
5.2 NUC on its custom-made daughter-board. .......................... 28
5.3 NUC’s custom-made motherboard. ................................... 28
5.4 Example testbed with four assembled NUC units. ............... 30
5.5 Current and voltage waveforms during the experiment. .......... 31
A.1 Schematic of connectors. .............................................. 40
A.2 Schematic of power stage. ............................................ 41
A.3 Schematic of signal isolation. ....................................... 42
A.4 Schematic of logic DC supplies. .................................... 43
A.5 Schematic of hot-swapping, stack initialization and protection. 44
A.6 Schematic of current sensing. ...................................... 45
A.7 Top layer of PCB. .................................................... 46
A.8 Ground layer of PCB. ................................................ 47
A.9 Signal layer of PCB. .................................................. 48
Chapter 1

INTRODUCTION

With the prevalence of the Internet as well as the increasing demands for cloud computing, data centers - the hosts of these services - are becoming more and more important. As the number and size of data centers continue to grow, so does the electrical power that they consume. In 2014, data centers in the U.S. consumed an estimated 70 billion kWh, which is approximately 1.8% of the total national electricity consumption that year [2]. With such a large energy usage, improving the energy efficiency of data centers has been a focus of research interest. The key component in data centers is the server, which operates at low DC voltage, typically 12 V or 48 V. Thus, a power delivery architecture is required to convert the utility AC voltage to low DC voltage to provide power for the servers. Increasing the efficiency of this power delivery architecture is a very crucial step in making data centers more energy efficient [3, 4, 5].

Current power delivery architectures for servers consist of cascaded power stages. As a typical example, a central rectifier draws from the utility AC voltage to regulate a DC bus voltage of 48 V. Then for each server, a DC-DC converter steps down the 48 V to the server’s nominal input 12 V voltage and supplies its power [5]. The server has numerous point-of-load converters to supply low voltages (e.g., 1 V, 3.3 V, etc.) to the various digital loads. Development of efficient and high-density converters for point-of-load conversion is an active area of research [6, 7, 8, 9, 10]. Each power conversion stage processes the full load power in this process. The system energy efficiency is directly limited by the efficiency of each power stage.

The series-stacked power delivery architectures are proposed to address the limitations of the conventional architectures, and can achieve much higher power delivery efficiency [11]. In the series-stacked architectures, the servers are connected in series, and differential power processing (DPP) converters can be used to compensate for the mismatch in the stacked servers’ currents.
Therefore, the bulk power consumed by the servers flows through the series-stack without being processed by the power conversion stage, and only the difference in power between servers is processed by the DPP converters. Using this technique, the amount of power processed, and the corresponding power loss, can be greatly reduced compared to conventional architectures, resulting in extremely high efficiency. There are three basic ways to connect the DPP converters: the server-to-server DPP architecture, server-to-bus architecture and server-to-virtual-bus architecture, which are depicted in Fig. 1.1(a), 1.1(b) and 1.1(c), respectively. The first architecture has been explored in [12, 13, 14], and the third architecture has been studied in [15, 16]. In [17] the hot-swapping operation of the servers is addressed in detail. There are also hybrid architectures combining the above mentioned types, as proposed in [11]. The DPP idea also applies to series-connected solar cells, where the first, second and third architectures are investigated in [18], [19] and [20], respectively. In this work, we focus on investigating the server-to-bus DPP architecture for data center servers.

It should be pointed out that for the series-stacked power delivery architecture, there are also research efforts that investigate using software only to compensate for the mismatch between the servers’ currents without using DPP converters [21]. This approach is not within the scope of this thesis.

The remainder of this thesis is organized as follows. Chapter 2 discusses the unique features of the server-to-bus power delivery architecture. In Chapter 3, two control methods for the DPP converters in the server-to-bus architecture are described. Chapter 4 introduces the prototype DPP hardware used in this work, and analyzes the power loss of the GaN-based DPP converter in detail. The experimental testbed of the architecture as well as the bench test results are presented in Chapter 5. In Chapter 6, the conclusion and future work are briefly stated.
Figure 1.1: Three DPP architectures.

(a) Server-to-server architecture.

(b) Server-to-bus architecture.

(c) Server-to-virtual-bus architecture.
Chapter 2

SERVER-TO-BUS DPP ARCHITECTURE

As can be seen in Fig. 1.1(b) and 1.1(c), the circuit connections of server-to-bus and server-to-virtual-bus architectures are very similar. The DPP converters are isolated DC-DC converters with one side connected to the DC bus (in the server-to-bus architecture shown in Fig. 1.1(b)) or a capacitor, virtual-bus (in the server-to-virtual-bus architecture shown in Fig. 1.1(c)), and the other side connected to the server.

The server-to-bus architecture has certain unique advantages compared with other DPP architectures. Firstly, it can achieve minimum total processed power in the DPP converters, and thus obtain the highest possible theoretical efficiency in power delivery [22]. Secondly, server-to-bus DPP architecture has a higher level of reliability than the other two architectures. These two points are explained below in greater detail.

2.1 Minimum Total Processed Power

The server-to-bus architecture can achieve the minimum total processed power in converters for most server load distributions in the series-stack. The other two DPP architectures cannot guarantee minimum total processed power in all server load distributions. To better understand this, let us assume that in a certain scenario, the servers are drawing current $i_{S1}$, $i_{S2}$, ... , $i_{Sn}$ as shown in Fig. 1.1(a), 1.1(b) and 1.1(c).

For the server-to-virtual-bus and server-to-server DPP architecture, each DPP converter’s output current $i_{DPPk}$ is uniquely determined (there is a only one feasible combination of the operating points of the DPP converters). The server-to-bus DPP architecture, however, is an under-determined system, where the DPP converters’ output currents have infinite feasible combinations. In fact, there are $n$ limiting variables ($i_{S1}$ through $i_{Sn}$), but $(n+1)$
controllable variables ($i_{DPP_1}$ through $i_{DPP_n}$, plus $i_{string}$), so there is one extra degree of freedom. For this architecture, string current $i_{string}$ can be set to any value as long as the DPP converters’ output currents ($i_{DPP_1}$ through $i_{DPP_n}$) are controlled correspondingly. Furthermore, some $i_{string}$ values will yield the combinations of $i_{DPP_1}$ through $i_{DPP_n}$ with smaller total processed power in the DPP converters than other combinations. We can thus control $i_{string}$ to be the optimal value(s) that results in the minimum total processed power in the converters of the server-to-bus DPP architecture. The following analysis will derive the value of the optimal string current in the server-to-bus architecture.

Let us assume, without loss of generality, that the server currents, as shown in Fig. 1.1(b), have the relationship

$$i_{S_1} \leq i_{S_2} \leq \cdots \leq i_{S_n}. \quad (2.1)$$

Moreover, for simplicity of analysis, assume 100% efficient converters. Then, the power processed in the $k$-th DPP converter $P_{DPP_k}$ is

$$P_{DPP_k} = |i_{DPP_k}| \cdot v_{Sk}, \quad (2.2)$$

where $v_{Sk}$ refers to the $k$-th server voltage. The objective to minimize, the total processed power in all the DPP converters $P_{DPP,\text{tot}}$, is

$$P_{DPP,\text{tot}} = \sum_{k=1}^{n} P_{DPP_k}. \quad (2.3)$$

If it is further assumed that the server voltages are all well regulated around the nominal input voltage, and thus

$$v_{S_1} \approx v_{S_2} \approx \cdots \approx v_{S_n} \approx V_{\text{nominal}}, \quad (2.4)$$

then

$$P_{DPP,\text{tot}} \approx V_{\text{nominal}} \sum_{k=1}^{n} |i_{DPP_k}|. \quad (2.5)$$

The new objective function to be minimized is then just the summation part $\sum_{k=1}^{n} |i_{DPP_k}|$, which is defined as $f(i_{string})$. 

5
\[ f(i_{\text{string}}) = \sum_{k=1}^{n} |i_{DPPk}| = \sum_{k=1}^{n} |i_{Sk} - i_{\text{string}}|. \] (2.6)

To determine the optimal \( i_{\text{string}} \) value that will minimize the objective function in (2.6), we will first simplify this expression and then look at its derivative. Suppose there are \( m \) servers (0 ≤ \( m \) ≤ \( n \)) whose currents are smaller than \( i_{\text{string}} \). Due to the assumption in (2.1), these servers are just server 1 to server \( m \). Then \( f(i_{\text{string}}) \) can be simplified as

\[ f(i_{\text{string}}) = \sum_{k=1}^{m} (i_{\text{string}} - i_{Sk}) + \sum_{k=m+1}^{n} (i_{Sk} - i_{\text{string}}), \] (2.7)

and the derivative is

\[ f'(i_{\text{string}}) = m - (n - m) = 2m - n. \] (2.8)

To better understand how \( f(i_{\text{string}}) \) changes with \( i_{\text{string}} \), two example cases with \( n \) being even (\( n = 6 \)) or odd (\( n = 7 \)) are considered, with the server currents in both cases listed in Tables 2.1 and 2.2. Example plots for \( f(i_{\text{string}}) \), \( m \) and \((2m - n)\) in the two example cases are shown in Fig. 2.1 and Fig. 2.2. It can be seen from the figures that in both cases, as \( i_{\text{string}} \) increases from 0, \( m \) increases monotonically from 0 to \( n \), and \((2m - n)\), which is also \( f'(i_{\text{string}}) \), increases monotonically from \((-n)\) to \( n \). Moreover, the objective function \( f(i_{\text{string}}) \) first decreases and then increases, and reaches the minimum (shown in red in the plots) when \((2m - n)\) crosses 0. When \( n = 6 \), the minimum happens when \( m = 3 \), when \( i_{S3} \leq i_{\text{string}} \leq i_{S4} \), or when \( i_{\text{string}} \) is larger than the currents of three (half of total) servers. In this case, there is a region of \( i_{\text{string}} \) values that are all optimal. When \( n = 7 \), the minimum happens when \( i_{\text{string}} = i_{S4} \), or when \( i_{\text{string}} \) equals to the server current right at the middle (the ‘median’). In general, when \( n \) is even, the optimal string current \( i_{\text{string,opt}} \) is any value between \( i_{S\frac{n}{2}} \) and \( i_{S(\frac{n}{2}+1)} \), and when \( n \) is odd, the optimal string current \( i_{\text{string,opt}} \) is equal to \( i_{S\frac{n+1}{2}} \), as expressed in Eq. 2.9. For a more detailed derivation of the optimal string current, readers can refer to [19].

It needs to be pointed out that the currents \( i_{Sk} \), \( i_{\text{string}} \) and voltages \( v_{Sk} \) are actually considered in terms of their averaged values in the analysis above. Due to the switching actions or operations in hysteresis mode of the power converters as well as the fast-changing nature of computation loads in servers,
the instantaneous values of these currents and voltages may fluctuate or ripple. These ripples are assumed to be small compared with the average values of these quantities within each control time period, so that their effects are neglected.

\[
    i_{\text{string, opt}} = \begin{cases} 
        i_{S_{n+1}}, & \text{if } n \text{ is odd.} \\
        \text{any value } \in [i_{S_{\frac{n}{2}}}, i_{S_{\frac{n}{2}+1}}], & \text{if } n \text{ is even.} 
    \end{cases} \tag{2.9}
\]

Table 2.1: Server currents in the example where n is even

<table>
<thead>
<tr>
<th></th>
<th>(i_{S_1}[\text{A}])</th>
<th>(i_{S_2}[\text{A}])</th>
<th>(i_{S_3}[\text{A}])</th>
<th>(i_{S_4}[\text{A}])</th>
<th>(i_{S_5}[\text{A}])</th>
<th>(i_{S_6}[\text{A}])</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2.2: Server currents in the example where n is odd

<table>
<thead>
<tr>
<th></th>
<th>(i_{S_1}[\text{A}])</th>
<th>(i_{S_2}[\text{A}])</th>
<th>(i_{S_3}[\text{A}])</th>
<th>(i_{S_4}[\text{A}])</th>
<th>(i_{S_5}[\text{A}])</th>
<th>(i_{S_6}[\text{A}])</th>
<th>(i_{S_7}[\text{A}])</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>
Figure 2.2: The $f(i_{\text{string}})$, $m$ and $(2m - n)$ in the example where $n$ is odd.

2.2 High Reliability

Another feature of the server-to-bus architecture is that it offers high reliability/availability compared with other DPP architectures. Since the server-to-bus architecture has one extra degree of freedom (one more control handle than quantities to be controlled), it can actually deliver power to $n$ servers with only $(n - 1)$ DPP converters. This indicates that it can tolerate one converter failure, and can still deliver power to all stacked servers. For example, if the DPP converter corresponding to the $k$-th server fails in the server-to-bus architecture, we can control the string current $i_{\text{string}}$ to be the $k$-th server’s current ($i_{S_k}$), and control the other DPP converters to inject or reject currents equal to the difference between this string current and the corresponding server currents [23]. For the other two DPP architectures, in the case of a single failure in the DPP converters, the power delivery can no longer be fulfilled.

As was quantitatively analyzed in detail in [23], due to the reduced power processed in the converters of the server-to-bus architecture, the reliability (characterized by down time over a 10 year time period) of the proposed series-stacked architecture is comparable to the reliability standard of a conventional power delivery architecture.
Chapter 3

CONTROL FOR SERVER-TO-BUS DPP ARCHITECTURE

In the series-stacked architectures, the DPP converters must be controlled properly to compensate for the mismatch between server loads. In this work, two control methods are used. The first one focuses on only the voltage regulation of the stacked servers which can be achieved by merely voltage sensing (no current sensing needed for control). One drawback is that the string current is not always optimized to achieve minimum power processed in the system. The second control method controls the string current to be optimal, and minimizes the total processed power. These two methods will be discussed in this chapter.

3.1 Voltage Hysteresis Control

The first control method used is bi-directional hysteresis control. A similar control method successfully achieved voltage regulation of stacked servers for the server-to-virtual-bus DPP architecture [1]. In this control algorithm, two hysteresis values $\varepsilon_0$ and $\varepsilon_1$ are pre-defined, as shown in Fig.3.1. If the server voltage goes higher than the reference voltage $V_{ref}$ plus $\varepsilon_1$, the DPP converter is controlled to deliver current from the server to bus direction, until the server voltage is lower than ($V_{ref} - \varepsilon_0$). After this the converter is shut down. If the server voltage goes lower than ($V_{ref} - \varepsilon_1$), the converter is controlled to deliver current from the bus to server direction, until the server voltage is higher than ($V_{ref} + \varepsilon_0$). After this the converter is shut down.

3.2 Optimal String Current Control

As was discussed in Chapter 2, only $(n-1)$ DPP converters are needed to keep the $n$ server voltages regulated in the server-to-bus architecture. The
control for optimal string current makes use of this feature. To control $i_{\text{string}}$ to be the optimal value as shown in Eq. 2.9, we can control the string current just to follow the ($\frac{n+1}{2}$)-th largest server current if $n$ is odd, or the ($\frac{n}{2} - 1$)-th largest server current if $n$ is even. To control the string current to follow a specific server’s current, we can turn off the single DPP converter corresponding to that specific server, and keep all other DPP converters operating as normal with the voltage hysteresis control stated in Section 3.1. In this control method, the server currents are measured, and then compared to determine which server consumes the ($\frac{n+1}{2}$)-th largest current (if $n$ is odd, or the ($\frac{n}{2} - 1$)-th largest if $n$ is even). Then the corresponding DPP converter is temporarily turned off. In this manner, the string current is controlled to be optimal in average, and the power processed in the DPP converters is minimized. In this work, a four-server system is investigated. Thus the optimal string current is any value between the second and third largest server current. In the optimal string current control method used in the experiments, the string current is controlled to follow the second largest server current.

Figure 3.1: Bi-directional hysteresis shape [1].
Chapter 4

PROTOTYPE DPP HARDWARE FOR THE SERVER-TO-BUS ARCHITECTURE

The DPP converter in the server-to-bus architecture, as shown in Fig. 1.1(b), is required to be bidirectional and isolated. For the experimental testbed of the four-server system in this work, the primary side of the converter is rated at 48 V, and the secondary side is rated at 12 V, which is the nominal server input voltage. The dual active bridge (DAB) topology is chosen for its symmetrical design and simple modulation [24]. In addition to the DPP converter, a stack initialization circuitry and hot-swapping circuitry are also needed. These three parts constitute the prototype DPP hardware used in the server-to-bus architecture of this work. The simplified schematic of the DPP hardware is shown in Fig. 4.1 and an annotated photograph of the prototype is shown in Fig. 4.2.

The stack initialization circuitry is used to achieve the voltage balancing between the series-stacked hardware prototypes when the DC bus is first applied to the series-stack. The hot-swapping circuitry provides complete isolation when the server is swapped out from the series-stack, and also limits the in-rush current caused by charging up the large input capacitor of the server when it is swapped in. Hot-swapping and initialization circuitries follow the principles explained in [17], which can also provide further details for interested readers. The following sections will discuss the operation characteristics of the DPP converter in greater detail.

4.1 Key Features of the GaN-Based DPP Converter Operation

This section discusses some key features in the operation of the DAB converter. GaN transistors are used in both the primary and secondary side full bridges, and the efficiency vs. output power plot of the converter is shown
Figure 4.1: The simplified schematic of the prototype DPP hardware.

Figure 4.2: The prototype DPP hardware.
Figure 4.3: The efficiency plot of the GaN-based DAB converter.

in Fig. 4.3. The key components are listed in Table A.1.

The zero voltage switching (ZVS) feature on all the eight switches, the impact of the secondary side parasitic inductance on the circuit, and the impact of the reverse conduction of the eGaN FETs during deadtime will be introduced below.

4.1.1 Zero Voltage Switching (ZVS)

One important feature of the DAB converter is that ZVS is achieved on all eight switches in normal operation. To understand this, let us first have a look at the converter operation in general. Shown in Fig. 4.4 is a detailed schematic of the DAB converter, and Fig. 4.5 shows the voltage and current waveforms of the converter under ideal conditions. One period of 5 $\mu$s is shown for the switching frequency $f_{sw}$ of 200 kHz. The converter is operated with simple phase shift modulation. The primary and secondary side full bridges generate two 50% duty ratio square waves shifted by a certain phase value $\phi$. The magnitudes of the two square waves are 48 V and 12 V, which are just the bus and server side voltages. With a transformer ratio of 4:1, $v_p$ transforms to a 12 V square wave $v'_p$. The voltage across the inductor is just $(v'_p - v_s)$, which results in an AC inductor current $i_L$ as shown in Fig. 4.5. The output current of the secondary side full bridge $i_{out}$ can be obtained by flipping half the period of the inductor current $i_L$ waveform when the switches S6 and S7 are on. Since $i_{out}$ is positive on average, power is delivered from
the bus side to the server side in this situation corresponding to Fig. 4.5.

Now let us focus on the switching intervals and see how ZVS is achieved.

Take the switching event occurring at $t = 0$ in Fig. 4.5 as an example, when
$S_1$ and $S_4$ are turned on, and $S_2$ and $S_3$ are turned off. Let us first take a
detailed look at the leg of $S_1$ and $S_2$ during this process, and the schematic
of this portion is in Fig. 4.6 with the $C_{DS}$ of each transistor shown. At $t = 0$,
$i_L$ is negative as can be seen in Fig. 4.5. With a 4:1 transformer, we have

$$i_{T1} \approx \frac{1}{4} i_L. \quad (4.1)$$

So $i_{T1}$ is also negative at $t = 0$. When switch $S_2$ is turned off, switch $S_1$
is not turned on until after the deadtime. During this deadtime period, the
negative $i_{T1}$ charges $C_2$ and discharges $C_1$. When $C_1$ is fully discharged,
$D_1$ starts to conduct the negative $i_{T1}$ current. And after the deadtime has
passed, switch $S_1$ is turned on with zero voltage across it, so ZVS turn-on is
achieved. As for the other pair of switches $S_3$ and $S_4$ in the other leg, the
situation is equivalent to this leg. The transformer current charges the $C_{DS}$
of $S_3$ while it turns off and discharges the $C_{DS}$ of $S_4$ during the deadtime,
and ZVS turn-on is achieved on $S_4$.

At $t = 2.5 \, \mu s$, when $S_1$ and $S_4$ are turned off, and $S_2$ and $S_3$ are turned
on, the situations for the two legs just flip each other - e.g. the leg of $S_1$ and
$S_2$ is what was the leg of $S_3$ and $S_4$ at $t = 0$, and thus ZVS is also true here.
From the analysis above, we can see that ZVS turn-on is achieved on all four
switches in the primary side.

In the secondary side, the same analysis can be carried out and we can see
that ZVS turn-on is still achieved in all the four switches. At the rising edge
Figure 4.5: Voltage and current waveforms of the DAB converter under ideal conditions.

Figure 4.6: Detailed schematic of the first leg of the primary side full bridge.
of the $v_s$ waveform in Fig. 4.5, secondary side switches S5 and S8 turn on, S6 and S7 turn off, and the inductor current $i_L$ is positive. Taking the leg of S5 and S6 as an example, $i_L$ charges the $C_{DS}$ of S6 while it turns off and discharges the $C_{DS}$ of S5 during the deadtime, and ZVS turn-on is achieved on S5. At the falling edge of the $v_s$ waveform in Fig. 4.5, ZVS turn-on is also true for both legs in the full bridge.

Therefore we can see that all eight switches have the feature of ZVS turn-on in the DAB converter in normal operation conditions where the above mentioned polarities of $i_L$ are guaranteed. The ZVS feature for the primary side switches significantly reduces the corresponding switching loss. For the secondary side, however, the benefits of ZVS are effectively not gained in the DAB topology, due to the high current of the secondary side together with the presence of parasitic inductance, as will be shown next.

4.1.2 Impact of Secondary Side Parasitic Inductance

Since the secondary side of the DAB converter is low voltage and high current, the presence of parasitic inductances that are in series with the switches can have large impact on converter operation. The parasitic inductances can be introduced by device packaging and PCB layout [25, 26, 24]. Shown in Fig. 4.7 is the schematic of one leg of the secondary side full bridge with the parasitic inductances included. The parasitic inductances cause the converter operation to be different from the ideal case shown in Fig. 4.5. Figure 4.8 is the oscilloscope screenshot of the DAB converter operation in a real experiment. As can be noticed, there are two places where the actual waveform differs greatly from the ideal one. One is the large overshoot of the $v_{leg3}$ waveform. The other one is the small stair in the voltage of the $v_{leg3}$ waveform every time before the rising or falling edges. These two phenomena are caused by the secondary side parasitic inductances that are in series with the switches. Let us first look into the second point, which is easier to understand.

The stair in voltage of the $v_{leg3}$ waveform happens when the inductor current $i_L$ charges fast, or when $\frac{di_L}{dt}$ has large magnitude. Take the voltage stair that happens before the first rising edge of the $v_{leg3}$ waveform for an example. Before the rising edge, switch S6 has been conducting. We then have:
\[ v_{\text{leg3}} = (L_{D,6} + L_{S,6}) \frac{\text{d}i_L}{\text{d}t}. \]  
(4.2)

If \( i_L \) is changing fast, the induced voltage on the parasitic inductances \( L_{D,6} \) and \( L_{S,6} \) can be large, causing \( v_{\text{leg3}} \) to be larger than zero, which is a stair in the waveform. Using the magnitude of the stair in the \( v_{\text{leg3}} \) waveform, we can actually calculate the parasitic inductance value as

\[ (L_{D,6} + L_{S,6}) = \frac{v_{\text{stair}}}{\frac{\text{d}i_L}{\text{d}t}}. \]  
(4.3)

Using \( v_{\text{stair}} = 1 \) V and \( \frac{\text{d}i_L}{\text{d}t} = 0.08633 \) A/ns as calculated from the waveform, we get \( (L_{D,6} + L_{S,6}) = 11.58 \) nH. We can further assume that \( (L_{D,i} + L_{S,i}) = 11.58 \) nH for \( i \) of 5, 7 and 8.

Next let us move on to the overshoot phenomenon in the \( v_{\text{leg3}} \) waveform. We can still take the first rising edge of the waveform as an example. At this rising edge, switch S6 turns off and switch S5 turns on, and the inductor current \( i_L \) is positive. Just before S6 turns off, the parasitic inductances \( (L_{D,6} + L_{S,6}) \) are conducting the same current as \( i_L \), and since \( i_L \) is quite large at this low-voltage high-current side, quite a large amount of energy
Figure 4.8: Oscilloscope screenshot of the DAB converter operation in a real experiment.

is stored in \((L_{D,6} + L_{S,6})\). When S6 turns off, the parasitic inductances \((L_{D,6} + L_{S,6})\) start to charge C6 and the current that they conduct starts to decrease. At the same time, the current starts to increase in the parasitic inductances \((L_{D,5} + L_{S,5})\) corresponding to S5. Since initial current and energy in \((L_{D,6} + L_{S,6})\) are quite large, resonance occurs between the parasitic inductances \((L_{D,6} + L_{S,6})\), \((L_{D,5} + L_{S,5})\) and the switch output capacitances C5 and C6, and large overshoot in voltage takes place. All the initial stored energy in the parasitic inductances is lost as the resonance is damped down by parasitic resistances in the resonant paths. After the deadtime, switch S5 is turned on, and the resonance settles to the steady state value. The switching loss in this process can be very large, and as is calculated in Section 4.2.2, it is the largest loss in the DAB converter in this operating condition compared to other losses such as conduction loss or core loss. A zoomed-in version of the rising edge in the \(v_{leg3}\) waveform is shown in Fig. 4.9, where the deadtime for the secondary side switches is 24 ns.
4.1.3 Impact of the reverse conduction of the eGaN FETs during deadtime

The switches of the DAB converter used are the eGaN FETs from EPC Inc. The eGaN transistors offer a lot of advantages over Si transistors, but one drawback of the eGaN switches is that the forward voltage drop of their ‘body diode’ can be around 2 V, and is higher than that of a Si-based FET. Actually eGaN transistors do not have a real body diode as in a Si MOSFET. It provides reverse conduction during deadtime through another mechanism, but the voltage drop for this reverse conduction is quite high. Therefore, for designs with the eGaN FETs, the deadtime needs to be optimized in order to shorten the reverse conduction time and obtain highest efficiency [27].

As shown in Fig. 4.9, following the rising edge of the $v_{\text{leg1}}$ waveform, there is a hump. This hump of approximately 2 V in magnitude corresponds to the reverse conduction of the primary side switch during deadtime. This deadtime of the primary side switches is 60 ns. For the secondary side, the effect of the reverse conduction during deadtime can be seen in the $v_{\text{leg3}}$ waveform in Fig. 4.9. In the ringing immediately after the rising edge, the steady state value of the voltage is actually about 14 V, which is approximately 2 V higher than the secondary side voltage of 12 V. After the deadtime, the steady state
Table 4.1: Resistances of the inductor, transformer windings, and the switches

<table>
<thead>
<tr>
<th>$R_L$ [mΩ]</th>
<th>$R_{Tr1}$ [mΩ]</th>
<th>$R_{Tr2}$ [mΩ]</th>
<th>$R_{S1}$ [mΩ]</th>
<th>$R_{S2}$ [mΩ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.18</td>
<td>14.4</td>
<td>2.1</td>
<td>3</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Table 4.2: Conduction loss of the inductor, transformer, and switches

<table>
<thead>
<tr>
<th>$P_{L,\text{cond}}$ [W]</th>
<th>$P_{Tr1,\text{cond}}$ [W]</th>
<th>$P_{Tr2,\text{cond}}$ [W]</th>
<th>$P_{S1,\text{cond}}$ [W]</th>
<th>$P_{S2,\text{cond}}$ [W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.021</td>
<td>0.106</td>
<td>0.247</td>
<td>0.044</td>
<td>0.42</td>
</tr>
</tbody>
</table>

value of the voltage returns to 12 V.

4.2 Power Loss Analysis

In this section, the power loss originating from different parts of the DAB converter is analyzed and calculated in detail. The operating condition corresponding to Fig. 4.8 with the phase shift value of 20° is chosen as the example in the loss calculation. In this case, power is delivered from the bus side to the server side with the input power $P_1$ being 127.3 W, output power $P_2$ being 119.1 W. The total power loss $P_{\text{loss, tot}}$ is 8.2 W, and converter efficiency is 93.5%. The switching frequency is 200 kHz.

4.2.1 Conduction Loss of Inductor, Transformer and Switches

The conduction loss of the inductor $P_{L,\text{cond}}$ can be calculated as

$$P_{L,\text{cond}} = I_{L,\text{rms}}^2 R_L,$$

where $I_{L,\text{rms}}$, the rms value of $i_L$ shown in Fig. 4.8, is calculated as 10.8377 A, and $R_L$, the resistance of the inductor winding, is listed in Table 4.1, together with the resistance values of the other components. The calculated $P_{L,\text{cond}}$ value is listed in Table 4.2, together with the conduction loss values of the other components. It should be pointed out that skin effect and other high-frequency effects are not taken into account in this calculation.

The conduction loss of the transformer primary and secondary windings...
$P_{Tr1,\text{cond}}$ and $P_{Tr2,\text{cond}}$ can be calculated as

$$P_{Tr1,\text{cond}} = \left(\frac{1}{4}I_{L,\text{rms}}\right)^2 R_{Tr1}$$ (4.5)

$$P_{Tr2,\text{cond}} = I_{L,\text{rms}}^2 R_{Tr2},$$ (4.6)

where $R_{Tr1}$ and $R_{Tr2}$ are the resistance of the primary and secondary windings of the transformer, respectively, with their values listed in Table 4.1. Again, skin and proximity effects are not included here. For the calculation of $P_{Tr1,\text{cond}}$ in Eq. 4.5, we assume that the current in the primary winding of the transformer is equal to one-fourth of the current in the secondary winding. This is assuming the magnetizing inductance of the transformer to be sufficiently large.

As for the conduction loss of the primary and secondary side switches, since two switches are connected to the transformer current in each half period, the losses are calculated as

$$P_{S1,\text{cond}} = \left(\frac{1}{4}I_{L,\text{rms}}\right)^2 \times 2 R_{S1}$$ (4.7)

$$P_{S2,\text{cond}} = I_{L,\text{rms}}^2 \times 2 R_{S2},$$ (4.8)

where $R_{S1}$ and $R_{S2}$ are the $R_{ds(\text{on})}$ of the primary and secondary side switches, respectively.

### 4.2.2 Switching Loss of the Switches

The switching loss is more complicated than the conduction loss. For the primary side, since ZVS is achieved, the normal switching loss can be neglected, and the only loss that needs to be considered is the loss from the reverse conduction voltage drop during deadtime. For the operating condition corresponding to Fig. 4.8, with the zoomed-in waveforms in Fig. 4.9, the duration $t_{r-c,1}$ of the primary side switch reverse conduction is 20 ns. The conduction current $I_{r-c,1}$ is approximately one-fourth of the inductor current at that moment, which is one-fourth of 8 A. With a reverse conduction voltage drop $V_{r-c,1}$ of 2 V, the energy lost in one leg of the full bridge during one switching event is
\[ E_{r-c,1} = t_{r-c,1} V_{r-c,1} I_{r-c,1} = 80 \text{ nJ}. \] \tag{4.9}

With a switching frequency of 200 kHz, and two switching event in one period for each leg of the full bridge, the power loss due to reverse conduction in the primary side switches is

\[ P_{r-c,1} = f_{sw} E_{r-c,1} \times 2 \times 2 = 0.064 \text{ W}. \] \tag{4.10}

Thus \( P_{S1,sw} = P_{r-c,1} = 0.064 \text{ W}. \)

For the secondary side, the switching loss is equal to power lost in reverse conduction during deadtime as well as in the parasitic inductance ringing phenomenon. For reverse conduction loss \( P_{r-c,2} \), with the duration of conduction \( t_{r-c,2} \) of 20 ns, conduction current \( I_{r-c,2} \) of 14 A, and a reverse conduction voltage drop \( V_{r-c,2} \) of 2 V, the energy lost in one leg of the full bridge during one switching event is

\[ E_{r-c,2} = t_{r-c,2} V_{r-c,2} I_{r-c,2} = 560 \text{ nJ}. \] \tag{4.11}

Thus

\[ P_{r-c,2} = f_{sw} E_{r-c,2} \times 2 \times 2 = 0.448 \text{ W}. \] \tag{4.12}

The energy loss from the parasitic inductances per leg per switching event \( E_{\text{par}} \) can be calculated as \cite{24}

\[ E_{\text{par}} = \frac{1}{2} \cdot L_{\text{series}} \cdot I_{\text{par}}^2 \cdot \frac{V_{pk}}{V_{pk} - V_{\text{secondary}}}, \] \tag{4.13}

where \( L_{\text{series}} = L_{D,5} + L_{S,5} + L_{D,6} + L_{S,6} \), and \( I_{\text{par}} \) is the current in the parasitic inductances when the switching event happens, which is 14 A. \( V_{pk} \) is a parameter given in \cite{24} as 32.7 V, and \( V_{\text{secondary}} \) is the secondary side voltage, which is 12 V. \( E_{\text{par}} \) is calculated as 3.5854 \( \mu \text{J} \). Therefore the power loss due to the secondary side parasitic inductances is

\[ P_{\text{par}} = f_{sw} E_{\text{par}} \times 2 \times 2 = 2.868 \text{ W}. \] \tag{4.14}

The secondary side switching loss is

\[ P_{\text{sw,2}} = P_{r-c,2} + P_{\text{par}} = 3.316 \text{ W}. \] \tag{4.15}
4.2.3 Core Loss of the Transformer and Inductor

The core loss of the transformer is determined by the curves given in the datasheet [28] of core loss vs $B_{pk}$. The peak flux density $B_{pk}$ is calculated using the formula in the datasheet

$$B_{pk} = \frac{KV_{\text{primary}}D_{\text{max}}}{f_{\text{sw}}N_{\text{primary}}},$$

(4.16)

where $K$ is the coefficient of $(58 \times 10^2)$, $D_{\text{max}}$ is the maximum duty cycle, which is 50% here, and $N_{\text{primary}}$ is the number of primary turns, which is 8. $B_{pk}$ is calculated as 87 mT, and the corresponding core loss $P_{T,\text{core}} = 0.3$ W.

The core loss of the inductor is calculated using the online inductor loss calculator provided by the manufacturer. The online calculator, however, only supports uni-directional inductor current, whereas the DAB converter inductor has bi-directional current. To still be able to obtain a proper core loss value, the inductor rms value setting in the online calculator is set as the actual rms value in the DAB converter, and the current ripple value setting is tuned until the inductor temperature matches the measured temperature value of 40.5 °C. The final core loss value obtained is $P_{L,\text{core}} = 0.15$ W.

4.2.4 Power Loss from Auxiliary Circuits

The power loss from auxiliary circuits includes the gate driving loss and the power consumption of the ICs for control. Another source for power loss is a resistive voltage divider for sensing the server side voltage for protection purposes. It has an approximately constant power consumption of 0.012 W. Since all the gate drivers and IC chips are supported by two LDOs (one for the Bus side, and the other for the Server side), the rest of the auxiliary loss is just the power delivered into the two LDOs. So we have

$$P_{\text{aux}} = P_{\text{vol-dvd}} + P_{\text{LDO,in,1}} + P_{\text{LDO,in,2}}.$$  

(4.17)

For LDO, its input and output current are basically equal, so its input power $P_{\text{LDO,in}}$ is

$$P_{\text{LDO,in}} = V_{\text{LDO,in}}I_{\text{LDO,out}}.$$  

(4.18)
and $I_{LDO,\text{out}}$ is just the sum of the current drawn by each of the downstream ICs.

As for the current drawn by IC chips, most IC datasheets give the typical current value that the chip draws. One special IC chip is the gate driver chip, as the current that it draws greatly depends on the switch that it drives. The gate driver chip input current is

$$I_{G-dr} = \frac{P_{G-dr}}{V_{DD,G-dr}},$$

(4.19)

where $V_{DD,G-dr} = 5$ V, and

$$P_{G-dr} = P_{G-dr-cap} + P_{\text{bootstrap}}.$$  

(4.20)

The $P_{G-dr-cap}$ is the power loss of charging and discharging the gate capacitance of the switch, and $P_{\text{bootstrap}}$ is the power loss of the bootstrap diode in the gate driver IC. We have

$$P_{G-dr-cap} = (Q_{G,\text{high-side}} + Q_{G,\text{low-side}})V_{DD,G-dr}f_{\text{sw}}.$$ 

(4.21)

And the $P_{\text{bootstrap}}$ is obtained from the curves provided in the datasheet.

The calculated values of the gate driver input currents are $I_{G-dr,1} = 15.6$ mA, $I_{G-dr,2} = 11.04$ mA. The other ICs draw currents of $I_{\text{other-IC},1} = 11.4$ mA, $I_{\text{other-IC},2} = 30.5$ mA. Thus

$$P_{LDO,\text{in},1} = 48 \text{ V} \times (2I_{G-dr,1} + I_{\text{other-IC},1}) = 2.0448 \text{ W}$$  

(4.22)

$$P_{LDO,\text{in},2} = 12 \text{ V} \times (2I_{G-dr,2} + I_{\text{other-IC},2}) = 0.631 \text{ W}.$$ 

(4.23)

And

$$P_{\text{aux}} = P_{\text{vol-dvd}} + P_{LDO,\text{in},1} + P_{LDO,\text{in},2} = 2.688 \text{ W}.$$ 

(4.24)

4.2.5 Summary of the Power Loss Analysis and Design Considerations for Converter Performance Optimization

From the above power loss analysis, we can get the loss breakdown as shown in Table 4.3. As can be seen, the power loss on the secondary side switches
Table 4.3: Loss breakdown of the DAB converter

<table>
<thead>
<tr>
<th>$P_L$ [W]</th>
<th>$P_{Tr}$ [W]</th>
<th>$P_{S1}$ [W]</th>
<th>$P_{S2}$ [W]</th>
<th>$P_{aux}$ [W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.171</td>
<td>0.653</td>
<td>0.108</td>
<td>3.736</td>
<td>2.688</td>
</tr>
</tbody>
</table>

is the largest. Inside this loss, the parasitic inductance generated loss makes up the majority. Therefore layout of the PCB that minimizes the parasitic inductances on the second side is one key step for converter efficiency optimization. Other than layout, advanced modulation schemes instead of the simple phase shift modulation may be employed to minimize the loss from parasitic inductances. The second largest loss in the table is the auxiliary circuit loss. This arises largely due to the use of high step-down LDOs. Future designs may consider using a switching converter to provide the 5 V logic power needed. All these above calculated losses add up to 7.356 W, which is not enough to match the measured value of 8.2 W. The difference may be due to the high-frequency effects that are not considered in the conduction losses of the inductor and transformer, and also may be due to the loss in the aluminum capacitor ESR, the loss in the current sensing wire for probing the inductor current, the loss in the RC filter for the PWM signals, and the loss from PCB trace resistance, etc. Future work may investigate these possibilities.
This chapter describes the experimental testbed of the server-to-bus DPP architecture and discusses the test results. These results verify the effectiveness of both control methods and the high efficiency of the server-to-bus DPP architecture.

5.1 Testbed

A testbed for the server-to-bus DPP power delivery architecture is developed with four Dell Optiplex SX280 servers running Linux operating systems. Each server has a single 12 V motherboard input, and the DC bus voltage for the four series-stacked system is 48 V, which is provided by a DC power supply (HP 6674A). The servers’ computational load is represented by running the standard Linux stress utility [29], or running the Hadoop software [30]. The voltage and current data in the system are sampled by a data acquisition unit (DAQ) from National Instruments at 5000 samples per second. The currents are converted to voltages with sense resistors and amplifiers (which are calibrated with an Agilent 34461A 6 1/2 digit multimeter) for the DAQ to sample. The control algorithm for the four DPP converters is implemented on a single off-board microcontroller (TI C2000 Piccolo F28069). Figure 5.1(a) and 5.1(b) show the schematic drawing and photo of the testbed respectively.

For future experimental work, a new testbed is planned based on a new model as the server - the Intel Next-Unit-of-Computing (NUC). NUC is a smaller sized yet more powerful server than the Dell Optiplex servers. The new server is just a single motherboard, and to connect it inside the series-stacked architecture, additional custom-made daughter-board and motherboard are designed and built, as shown in Fig. 5.2 and 5.3 respectively. The
(a) The schematic of the testbed for the server-to-bus architecture.

(b) The photo of the testbed for the server-to-bus architecture.

Figure 5.1: The testbed for the server-to-bus architecture.
5.2 Experimental Results

The experiment to test the server-to-bus architecture takes 560 seconds in total, where the servers initialize, operate and execute computation tasks and finally shut down. When the servers are in operation, the DPP converters are controlled using both of the above mentioned control methods. These different situations correspond to nine different time intervals, as shown in the

daughter-board and motherboard are equipped with PCIe edge connectors through which they are assembled together. An example testbed with four assembled NUC units is shown Fig. 5.4.
Table 5.1: Breakdown of the average input and output powers, the average power loss, the system-level efficiency and power conversion efficiency during the experiment

<table>
<thead>
<tr>
<th>Time Interval [s]</th>
<th>Interval 1</th>
<th>Interval 2</th>
<th>Interval 3</th>
<th>Interval 4</th>
<th>Interval 5</th>
<th>Interval 6</th>
<th>Interval 7</th>
<th>Interval 8</th>
<th>Interval 9</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt; P_{in} &gt; ) [W]</td>
<td>279.59</td>
<td>462.85</td>
<td>479.69</td>
<td>484.96</td>
<td>252.43</td>
<td>418.04</td>
<td>422.88</td>
<td>421.93</td>
<td>243.71</td>
<td>361.29</td>
</tr>
<tr>
<td>(&lt; P_{out} &gt; ) [W]</td>
<td>268.71</td>
<td>451.43</td>
<td>467.35</td>
<td>472.42</td>
<td>242.47</td>
<td>395.72</td>
<td>401.88</td>
<td>398.62</td>
<td>232.64</td>
<td>347.15</td>
</tr>
<tr>
<td>(&lt; P_{loss} &gt; ) [W]</td>
<td>10.87</td>
<td>11.42</td>
<td>12.34</td>
<td>12.54</td>
<td>9.96</td>
<td>22.31</td>
<td>21.01</td>
<td>23.31</td>
<td>11.07</td>
<td>14.14</td>
</tr>
<tr>
<td>(\eta_{sys} \ [%])</td>
<td>96.31</td>
<td>97.83</td>
<td>97.73</td>
<td>97.72</td>
<td>96.22</td>
<td>94.94</td>
<td>95.34</td>
<td>94.75</td>
<td>95.64</td>
<td>96.35</td>
</tr>
<tr>
<td>(\eta_{conv} \ [%])</td>
<td>97.21</td>
<td>98.99</td>
<td>98.94</td>
<td>98.95</td>
<td>96.97</td>
<td>96.04</td>
<td>96.70</td>
<td>95.85</td>
<td>96.43</td>
<td>97.42</td>
</tr>
<tr>
<td>( &lt; i_{S1} &gt; ) [A]</td>
<td>6.06</td>
<td>9.49</td>
<td>9.87</td>
<td>9.93</td>
<td>5.65</td>
<td>9.80</td>
<td>9.91</td>
<td>9.87</td>
<td>5.20</td>
<td>7.88</td>
</tr>
<tr>
<td>( &lt; i_{S2} &gt; ) [A]</td>
<td>6.13</td>
<td>9.48</td>
<td>9.88</td>
<td>9.97</td>
<td>5.72</td>
<td>9.84</td>
<td>10.16</td>
<td>9.92</td>
<td>5.16</td>
<td>7.92</td>
</tr>
<tr>
<td>( &lt; i_{S3} &gt; ) [A]</td>
<td>5.47</td>
<td>9.48</td>
<td>9.71</td>
<td>9.84</td>
<td>4.14</td>
<td>3.80</td>
<td>3.84</td>
<td>3.75</td>
<td>4.94</td>
<td>5.97</td>
</tr>
<tr>
<td>( &lt; i_{S4} &gt; ) [A]</td>
<td>4.97</td>
<td>9.69</td>
<td>10.05</td>
<td>10.21</td>
<td>4.86</td>
<td>10.02</td>
<td>10.21</td>
<td>10.17</td>
<td>4.28</td>
<td>7.55</td>
</tr>
<tr>
<td>( &lt; i_{string} &gt; ) [A]</td>
<td>6.17</td>
<td>9.59</td>
<td>9.94</td>
<td>10.00</td>
<td>5.59</td>
<td>8.77</td>
<td>10.27</td>
<td>8.73</td>
<td>5.40</td>
<td>7.82</td>
</tr>
</tbody>
</table>
detailed current and voltage waveforms in Fig. 5.5. Based on the voltage and current data, the input power to the system $P_{in}$ is calculated by multiplying the bus voltage and current, and the output power $P_{out}$ is calculated by summing the multiplication of the individual server voltages and currents. The power loss $P_{loss}$ is the difference between $P_{in}$ and $P_{out}$.

$$P_{loss} = P_{in} - P_{out}. \quad (5.1)$$

There are various sources for the power loss in the system. Firstly, there is the power loss from the DPP converters, $P_{loss,conv}$. Apart from this, there are power losses from the current sensing resistors, $P_{loss,meas}$; from system wiring, $P_{loss,wiring}$; from the hot-swapping switches, $P_{loss,HS}$, etc. Based on these understandings, two types of power delivery efficiencies are calculated: One is system-level efficiency, $\eta_{sys}$, where the loss from current sensing resistors is excluded,

$$\eta_{sys} = 1 - \frac{P_{loss} - P_{loss,meas}}{P_{in}}. \quad (5.2)$$

The other is power conversion efficiency, $\eta_{conv}$, where the power losses from current sensing resistors, system wiring and the hot-swapping switches are
Figure 5.5: Current and voltage waveforms during the experiment.
excluded,

\[ \eta_{\text{conv}} = 1 - \frac{P_{\text{loss}} - P_{\text{loss,meas}} - P_{\text{loss,wiring}} - P_{\text{loss,HS}}}{P_{in}}. \]  

(5.3)

A similar calculation is explained in detail in [17]. A breakdown of the average input and output powers, the average power loss, the system-level efficiency \( \eta_{\text{sys}} \) and power conversion efficiency \( \eta_{\text{conv}} \) is given in Table 5.1, and the average current of each server during each interval is also listed at the bottom of the table.

As can be seen in Fig. 5.5, all server voltages are regulated to their nominal 12 V throughout the entire test. This verifies the effectiveness of both control methods for the DPP converters. In the first 125-second time interval, the servers are initialized. The loads are not so balanced during this period, and the series-stacked system has not yet achieved its highest efficiency. The system-level efficiency is calculated as 96.31\% and the power conversion efficiency is calculated as 97.21\%. Then the next three time intervals (Intervals 2, 3 and 4) represent the scenario when the server loads are quite balanced, which is very advantageous for the series-stacked power delivery architecture. Here the four servers are all running the Linux Stress computation tasks for altogether 150 s. In the first 60 s (Interval 2) and last 30 s (Interval 4), the DPP converters are controlled with the normal voltage hysteresis control, and in the second 60 s (Interval 3, shaded area in Fig. 5.5), the converters are controlled with the optimal string current control. The highest efficiencies are achieved here, with the power conversion efficiency reaching 98.99\% and system-level efficiency reaching 97.83\% in Interval 2. One may notice that the efficiencies when the optimal string current control is employed are slightly lower than those with the normal voltage hysteresis control. Actually, when the four server loads are very balanced (their currents very close), the string current will settle at the near-optimal region regardless of whether one uses the normal voltage hysteresis control or the optimal-string current control method. The advantage of the optimal string current control is not apparent in this scenario, and its efficiency may be deteriorated by the larger voltage ripples and thus larger losses in the stack-side capacitors when trying to regulate the four server voltages using just three DPP converters in this control method. The benefits brought by the optimized string current become evident when the server loads are quite unbalanced - when the string
current with the normal control method may not settle at the optimal value. This scenario is represented by Intervals 6, 7 and 8, when three servers (server 1, 2 and 4) are being stressed while one server is idle (server 3). For the first 60 s (Interval 6) and last 30 s (Interval 8), the DPP converters are controlled with the voltage hysteresis control, and for the second 60 s (Interval 7, shaded area in Fig. 5.5), the converters are controlled with the optimal string current control. Now it can be seen that the efficiencies corresponding to the optimal string current control are higher than with the voltage hysteresis control method. There is a 0.66% boost in power conversion efficiency when changing from the normal hysteresis control (Interval 6) to the optimal string current control (Interval 7), and a 0.85% efficiency drop when changing from the optimal string current control (Interval 7) back to the normal hysteresis control (Interval 8). Also it can be noted that the string current rises from 8.77 A to 10.27 A when changing from the normal voltage hysteresis control to the optimal string current control, where the average string current \( \langle i_{\text{string}} \rangle \) is effectively equal to the second largest server current \( i_{S2} \) \( \langle i_{\text{string}} \rangle \) is slightly larger as shown in Table 5.1 because a small portion of it also flows to power the logic circuits in the DPP converters). After these three intervals, the servers are shut down in Interval 9. Interval 5 is an interval between the two long computation tasks, during which the four servers are idle. Over the entire experiment period, the system-level efficiency and power conversion efficiency are calculated as 96.35% and 97.42%, respectively.
Chapter 6

CONCLUSION

This thesis presents the first proof-of-concept experimental demonstration of a fully operational server-to-bus DPP architecture for data center power delivery. The server-to-bus DPP architecture has its unique advantages of being able to achieve the minimum power processed in the DPP converters, as well as having higher reliability than other DPP architectures. DAB converters with GaN switches are used as the DPP converters in the prototype. 98.99% efficiency is achieved while the servers are running real-life data center operation. This work also shows with experiments that the server-to-bus DPP architecture can deliver power to $n$ servers with only $(n-1)$ DPP converters in operation, which implies that the server-to-bus architecture can tolerate one converter failure. Future work will include the hot-swapping operations of the servers as well as optimization of the system to further improve power delivery efficiency.
References


Appendix A

PROTOTYPE DPP HARDWARE
COMPONENT LIST AND PCB LAYOUT

Table A.1: The key components of the DPP converter

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaN Switches</td>
<td>EPC 2032 (Bus side)</td>
</tr>
<tr>
<td></td>
<td>EPC 2030 (Server side)</td>
</tr>
<tr>
<td>GaN Gate Drivers</td>
<td>TI LM5113</td>
</tr>
<tr>
<td>Transformer</td>
<td>Coilcraft PL300-100L, 8:2 turns</td>
</tr>
<tr>
<td>Inductor</td>
<td>Coilcraft SLC1480-201ML, 200 nH</td>
</tr>
<tr>
<td>$C_{in}$</td>
<td>ceramics 8 × 1 $\mu$F, TDK</td>
</tr>
<tr>
<td></td>
<td>aluminum 2 × 68 $\mu$F SMD, Panasonic</td>
</tr>
<tr>
<td>$C_{out}$</td>
<td>ceramics 8 × 4.7 $\mu$F, TDK</td>
</tr>
<tr>
<td></td>
<td>aluminum 2 × 1 mF SMD, Panasonic</td>
</tr>
</tbody>
</table>
Figure A.1: Schematic of connectors.
Figure A.2: Schematic of power stage.
Figure A.3: Schematic of signal isolation.
Figure A.4: Schematic of logic DC supplies.
Figure A.5: Schematic of hot-swapping, stack initialization and protection.
The output amplifier has a low impedance output and is designed to drive up to 200pF capacitive loads directly. Capacitive loads exceeding 200pF should be decoupled with an external resistor of at least 100Ω.

Figure A.6: Schematic of current sensing.
Figure A.7: Top layer of PCB.
Figure A.8: Ground layer of PCB.
Figure A.9: Signal layer of PCB.
Figure A.10: Bottom layer of PCB.
Appendix B

KEY MICROCONTROLLER CODE

```c
// Sampling stack voltages and server currents

interrupt void adc_isr(void)
{
    // this happens @ every switching cycle (= 5mus period)

    if (adc_count_V >= (sampling_time_V - adc_divider_V)) // last
        n measurements, this if takes ~300ns
    {
        V4[adc_i_V] = AdcResult.ADCRESULT1; // V4 / 6
        V3[adc_i_V] = AdcResult.ADCRESULT2; // V3 / 12
        V2[adc_i_V] = AdcResult.ADCRESULT3; // V2 / 18
        V1[adc_i_V] = AdcResult.ADCRESULT4; // V1 / 24
        adc_i_V += 1;
    }
    if (adc_count_I == 399) // every 2000mus
    {
        I4[adc_i_I1] = AdcResult.ADCRESULT0;
        I3[adc_i_I1] = AdcResult.ADCRESULT5;
        I2[adc_i_I1] = AdcResult.ADCRESULT6;
        I1[adc_i_I1] = AdcResult.ADCRESULT7;
        adc_i_I1 += 1;
        adc_count_I = 0;
    }
    else
    {
        adc_count_I += 1;
    }
    if (adc_i_I1 == 10) // collected for 20ms
    {
        I1_first_sum[adc_i_I2] = 0;
        I2_first_sum[adc_i_I2] = 0;
        I3_first_sum[adc_i_I2] = 0;
    }
}
```
I4.first_sum[adc_i.I_2] = 0;
for (adc_i.I_1 = 0; adc_i.I_1 < 10; adc_i.I_1++)
{
    I1.first_sum[adc_i.I_2] += I1[adc_i.I_1];
    I2.first_sum[adc_i.I_2] += I2[adc_i.I_1];
    I3.first_sum[adc_i.I_2] += I3[adc_i.I_1];
    I4.first_sum[adc_i.I_2] += I4[adc_i.I_1];
}
adc_i.I_2 += 1;
adc_i.I_1 = 0;
}
if (adc_i.I_2 == 50) // reached the i control period of 1s
{
    adc_i.ready_flag = 1;
    adc_i.I_2 = 0;
}
if (adc_count_V == sampling_time_V - 1) // reached the v control period of 500mus
{
    GpioDataRegs.GPBToggle.bit.GPIO33 = 1;
    adc_V.ready_flag = 1;
    adc_count_V = 0;
}
else
{
    AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; // Clear ADCINT1 flag reinitialize for next SOC
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; // Acknowledge interrupt to PIE
    adc_count_V += 1;
}
return;

// Determining which DPP converter to be temporarily turned off
void Determine_Off_Dpp()
{
    I_s1 = 0;
    I_s2 = 0;
    I_s3 = 0;
I.s4 = 0;

for (adc_i.I.2 = 0; adc_i.I.2 < 50; adc_i.I.2++)
{
    I.s1 += (I1_first_sum[adc_i.I.2]);
    I.s2 += (I2_first_sum[adc_i.I.2]);
    I.s3 += (I3_first_sum[adc_i.I.2]);
    I.s4 += (I4_first_sum[adc_i.I.2]);
}

I.sort[0] = I.s1;
I.sort[1] = I.s2;
I.sort[2] = I.s3;

for (j = 0; j < 2; j++)
{
    for (k = 0; k < (3 - j); k++)
    {
        if (I.sort[k] > I.sort[k+1])
        {
            swap = I.sort[k];
            I.sort[k] = I.sort[k+1];
            I.sort[k+1] = swap;
        }
    }
}

if (I.s1 == I.sort[2])
    off_dpp = 1;
if (I.s2 == I.sort[2])
    off_dpp = 2;
if (I.s3 == I.sort[2])
    off_dpp = 3;
if (I.s4 == I.sort[2])
    off_dpp = 4;
adc_i.I.2 = 0;

mcu_code.c