Getting Started With *DCDCbuck* Board

Prof. Dr. Martin J. W. Schubert, Electronics Laboratory,

OTH Regensburg, Regensburg, Germany

Getting Started With *DCDCbuck* Board

Abstract. This communication presents the *DCDCbuck* converter board, which is used as daughter board to *Terasic's DE1-SoC* board.

1 Introduction

1.1.1 Objectives

This document introduces to the usage of the *DCDCbuck* board, performing DC/DC step-down conversion as daughter board of Terasic's *DE1-SoC* board. With an A/D converter available, the *DCDCbuck* board can also be used with other *DEx* boards with compatible user header, e.g. *DE2, DE2-70, DE2-115.*

1.2 Assumed Test Environment

1.2.1 *DE1-SoC* Hardware

This document assumes that you are familiar with the *Terasic's* [1] *DE1-SoC* board [2] or a similar *DEx* board with the same general-purpose input/output (GPIO) user header. The version of your *DE1-SoC* board can be identified at [3]. *DE1-SoC* board revisions *F* and *G* differ in a printed company label only. To get this information from the internet, navigate your browser to [4] and download *DE1-SoC_v.5.1.2_HWrevF_SystemCD.zip* [5] or a later version. It contains amongst other things important documents such as *DE1-SoC User Manual* [6] and *Schematic* [7]. On the computer system of OTH Regensburg you will also find the CD on drive K:\Sb\ [8]. Do not use any other manual revision to follow this documentation. The differences may be considerable.

1.2.2 *Quartus II* [9] and *ModelSim* [10] Software Tools

It is assumed that you have *Intel's ModelSim [10] and Quartus II 18 [9]* software available. To download this freeware for your private PC you have to sign in at *Intel* [11]. At OTH Regensburg's PC pools of faculties *EI* and *IM,* this software is installed. At faculty *EI,* also *Quartus II 8* is installed supporting the older *DE2* boards with *Cyclone II* FPGAs, which are not supported for *Quartus II* versions greater than 13.1.

1.2.3 Use of *VHDL*

The *IEEE standard VHDL Language reference manual* [12] is difficult to read. *Qualis VHDL Quick Reference Card* [13] and *1164 Packages Quick Reference Card* [14] are compact but difficult to understand. Feel free to find your own sources.

VHDL is not case sensitive. In the following, **`VHDL KEYWORDS`** will be written in **`ALL CAPITAL LETTERS`** and user defined names in **`lowercase letters`**. Exception are capitalized initials

used for composed self-made names, e.g. *AddressBus* or *DataBusBitWith*. Self-made data types begin with *t_,* e.g. *t_StateVector*.

1.3 Contents

The organization of this document is as follows:

Section **1** is this **introduction**,
Section **2** presents the ***DCDCbuck*** board **hardware**,
Section **3** suggests some **tests and basic characterization** methods of the *DCDCbuck* board,
Section **4** draws relevant **conclusion** and
Section **5** offers **references.**

1.4 Acknowledgements

The author would like to thank *Terasic Technologies* [1] for authorization to use copies of *Terasic* documentation for *DE1-SoC* boards for teaching purposes.

2 The *DCDCbuck* Daughter Board Hardware

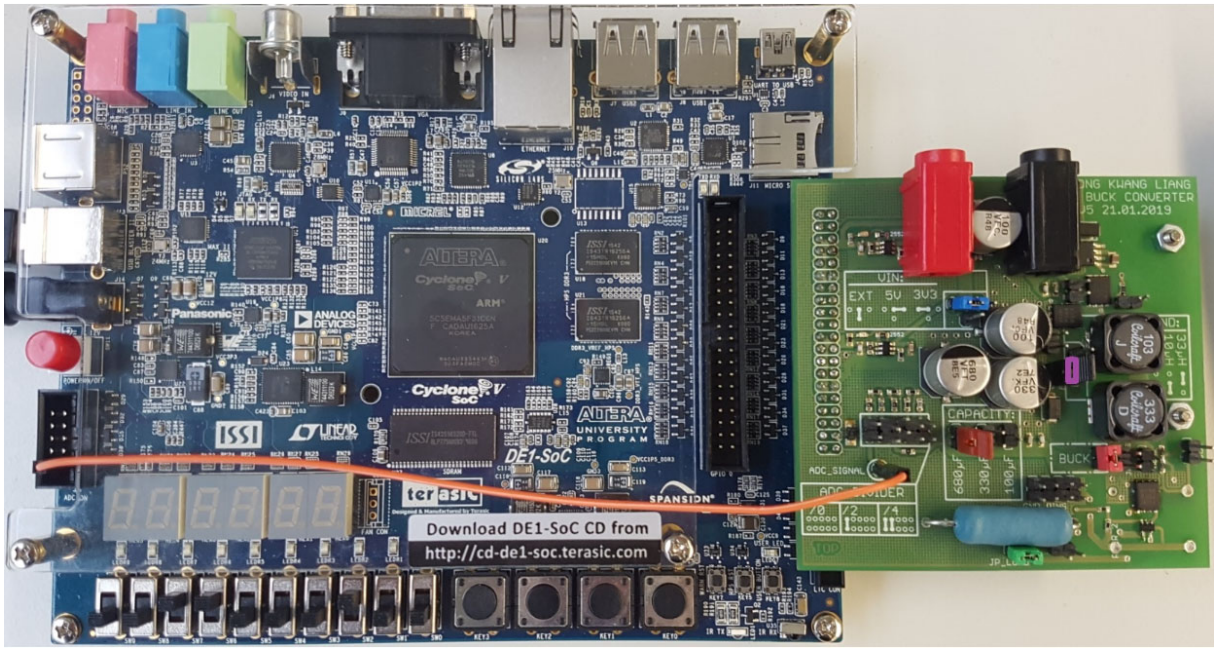


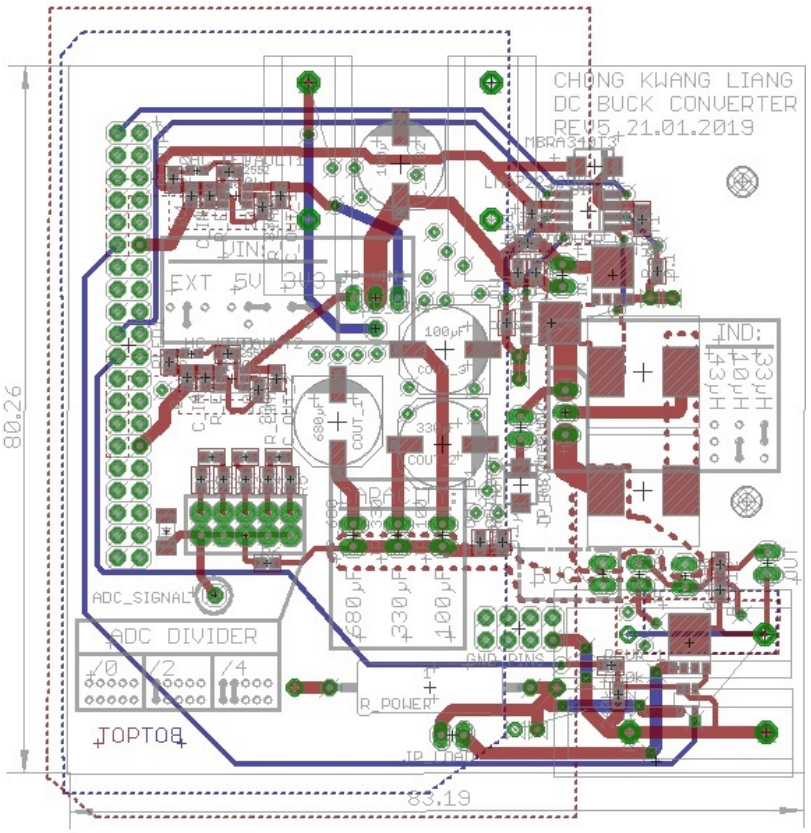
Fig. 2.1: *DE1-SoC* board (left) carrying the *DCDCbuck* daughter board (right). The orange wire connects the input of the *DE1-SoC* on-board's ADC (*LTC2308*) channel 0 with the *DCDCbuck* board's output voltage.

Fig. 2.2:
Eagle [18] Layout of *DCDCbuck* board.

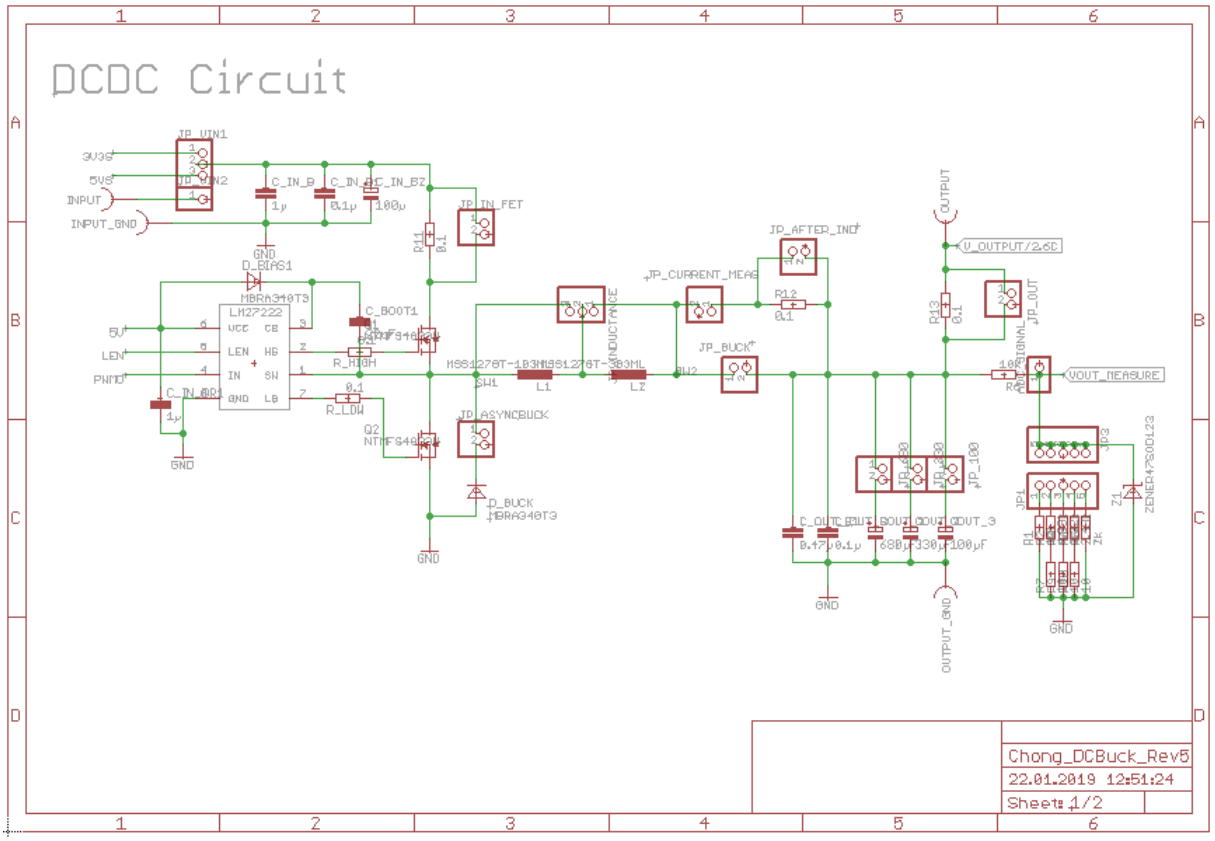
Brown: top layer metal wires

Blue: bottom layer metal wires

Green: vias & contacts.



(a) View 1: DC/DC buck converter



(b) View 2: 5V and 3.3V input current limiters, user and ground header, load current circuit

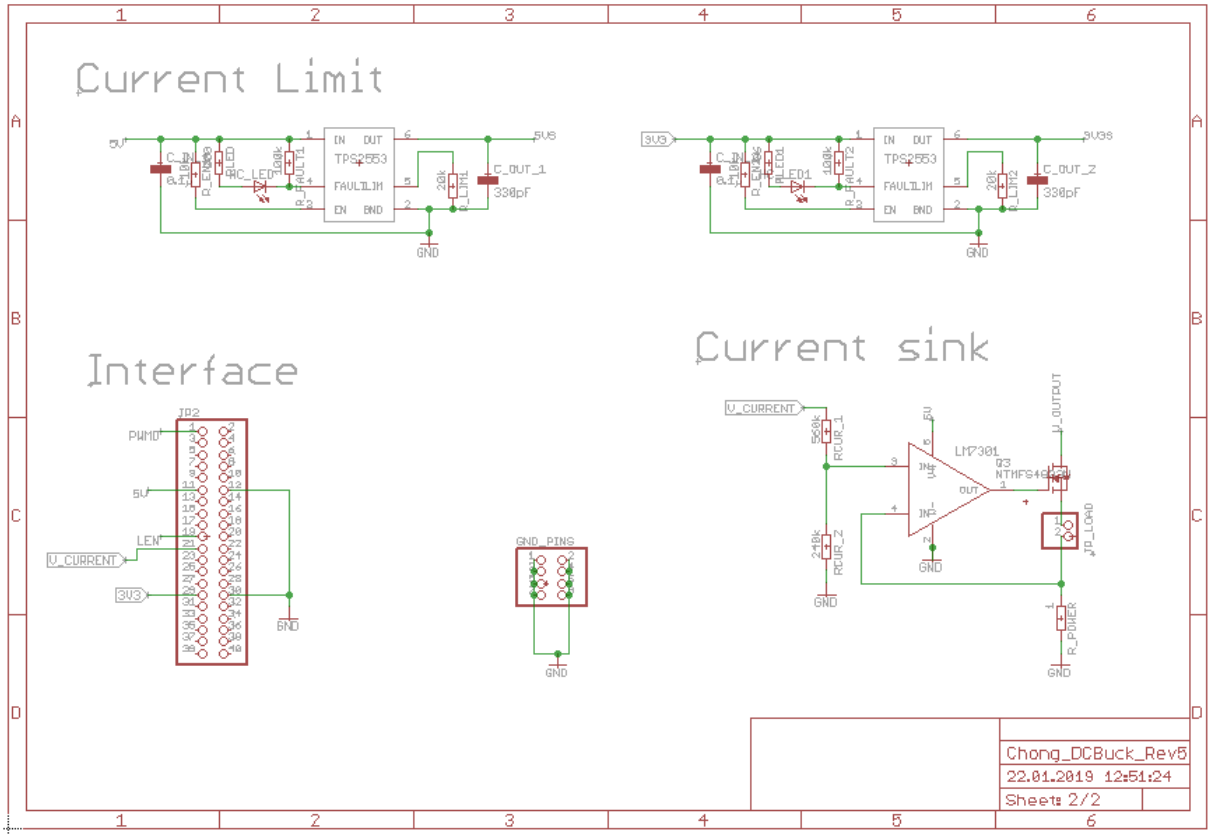


Fig. 2.3: *Eagle* [18] schematic views of the *DCDCbuck Rev. 5* board, by K.-L- Chong [19].

Fig. 2.1 shows the *DE1-SoC* board (left) with plugged-in *DCDCbuck* daughter board (right). The orange wire connects the *DCDCbuck* board's output voltage with the ADC *LTC2308*'s input channel 0 (lower left pin), whereas the *LTC2308* is on board of the *DE1-SoC*.

Fig. 2.2 illustrates the layout of the *DCDCbuck* board with top metal layer brown, bottom metal layer blue and contacts / vias green. (Vias connect different metal layers.)

Fig. 2.3 is the schematic part of (a) the main circuit, and (b) 3.3V and 5V current limiter, interface to *DE1-SoC* board and current sink.

3 *DCDCbuck* Board

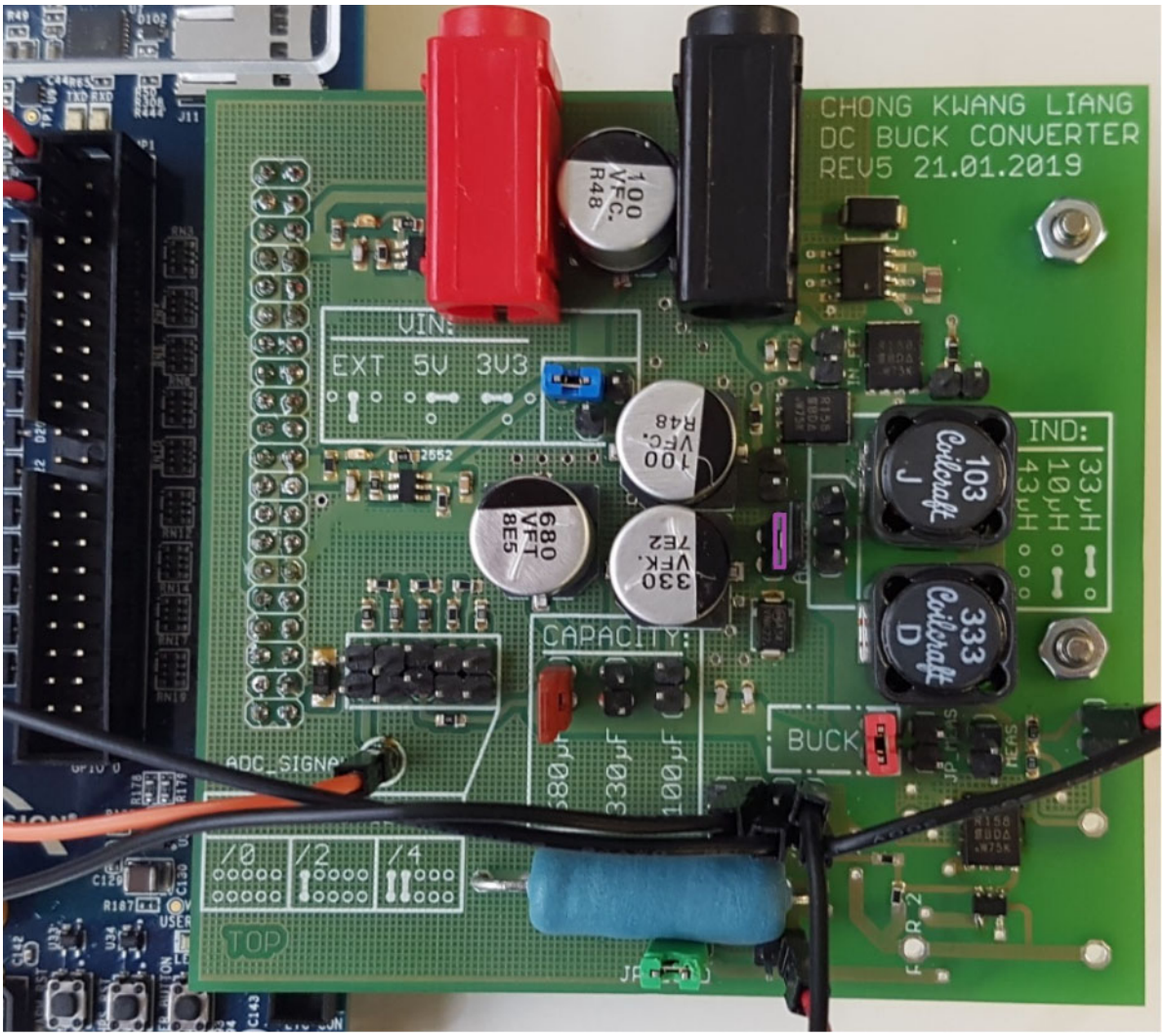


Fig. 3.0: *DCDCbuck* board setup for first tests.

3.1 Starting *DCDCbuck* Board

Starting with *DE1-SoC* and *DCDCbuck* Hardware
Plug daughterboard *DCDCbuck* into user header *JP2* (=*gpio*_1) of *DE1-SoC* board (Fig. 3.0).

On the *DCDCbuck* daughter board:

- Connect cable (orange in Figs. 2.1 and 3.0) from *DCDCbuck* board's pin *ADC_SIGNAL* to *DE1-SoC* board's user header *ADC CON*, left lower pin (Fig. 2.1).
- Set jumper 3V3 (blue in Fig. 3.0) to supply the daughter board with 3.3V. If a red LED lights up, then the current limiter was activated by a current peak. You will have to unplug the DCDCbuck daughter board or switch off *DE1-SoC* board to reset the current limiter.
- Set jumper *JP_Buck* (red in Fig. 3.0) to get a current path to the output.
- Set jumper *JP_Load* (green in Fig. 3.0) to enable the load current circuitry.
- Set jumper 680uF (brown in Fig. 3.0) to connect the load capacitor
- No need but nice to have for better efficiency: Set jumper *Async* (pink in Fig. 3.0).

Starting with *Quartus II* Software
Use *Quartus II 18* or later to program *DE1-SoC* board with file *ci_de1soc_DCDCbuck.sof*, that you will get from your supervisor.

- Set switches of the *DE1-SoC* board to *sw*(9:0) = "0101 00 11 00".
- Get file *ci_de1soc_DCDCbuck.zip* and extract it into directory …*ci_de1soc_DCDCbuck*. Contains is subdirectory .*output* with files **.cdf* and **.sof*.
- Within directory …*ci_de1soc_DCDCbuck* click on file *ci_de1soc_DCDCbuck.qpf*. *Quartus II 18* (or later) should start with the actual directory as working directory.
- Select from *Quartus II* main menu: *Tools* > *Programmer*. The programmer window should open with *Hardware setup…* window showing string *DE1-SoC [USB-1]*. Otherwise, look at *Getting Started with DE1-SoC Board* from the author or any other source to get it running.
- Click on programmer's button *Start* to program the FPGA. You should see a reaction at the LEDs.

Starting the Board
On the 7seg- display, you should see something like "`i 1200`", which is the set point in millivolts (mV). Set *sw*(3) = 0. You should now see output voltage the measured by the ADC indicated on the 7seg display like "`U_1xxx`" with *xxx* being close to 200.

Got it running? Congratulations! You got the board working!

Using the Oscilloscope

Look at the author's document *Quickstarting_DSoX2024_Oscilloscope.pdf* to handle the *DSO-X2024* oscilloscope. In the following, to connect "channel #" stand for: connect the red end of the BNC cable to # and the black end to ground. As can be seen from Fig. 2.1, there is a group of 2x4 ground pins to the right of label "100 µF" near to the big blue load resistor. In Fig. 3.0 four black ends of BNC-cables are plugged onto these 8 pins, so that some of them cannot be seen in this figure.

Interconnections to the *DSO-X 2024* oscilloscope:

- PWM signal: Connect oscilloscope's channel 1 (CH1, yellow) to *pin 1* (upper left) of user header *gpio_0*. (We use *JP1==gpio_0*, as *JP2==gpio_1* is occupied by the DCDCbuck board. *VHDL* statement "*gpio0 <= gpio_1*" copies user header *gpio_1* to *gpio_0*.)
- Load: Connect CH2 (green) of the oscilloscope to the right jumper pin on the right side of jumper JP_LOAD (green). I shows the voltage at the 1Ω load resistor, typically 0 or 1V.
- Connect CH3 (blue) of the oscilloscope to pin 3 (below pin 1, see -> Fig. 3.0) of user header *gpio_0* of the *DE1-SoC* board.
- Output voltage: Connect CH4 (red) of the oscilloscope to pin labeled *OUT* of the *DCDCbuck* board.

Oscilloscope *Auto Scale*

- Switch CH1, CH2, CH3, CH4 of the oscilloscope on. Then press hardkey *Auto Scale*.
- The screen will show an image similar to that of Fig. 3.1(a).

Triggering

- The oscilloscope triggers now the rising edge of CH1 (yellow), indicated by the little yellow triangles at the top and left edge of the screen.
- CH1 (yellow) shows the output of the pulse-width modulating (PWM) DAC.
- Push switches *sw*(9:6) to "1111". Then the PWM-DAC outputs 1's only and the oscilloscope has no more edge to trigger.
- Change the setting of your oscilloscope such, that it triggers CH3 (blue). Now you should have a fixed scope image again.
- It is not wise to scarify a valuable input channel for a simple trigger signal. Consequently:
- Disconnect the BNC cable from CH3, plug it at the backside of the scope to external trigger in (*EXT TRIG IN*).
- Switch OFF CH3 of the scope (pressing 1 or 2 times on the respective button at the scope).
- At your scope, select trigger channel external (*EXT*).
- Press HK *AutoScale*. You should now get a standing image on the oscilloscope again.
- Set *sw*(9:6)=1000 and Press *AutoScale*. You should get an image similar to Fig. 3.1(b).

Observing Some Effects

Disconnect all 4 black ground ends from the measurement cables from the ground pins on the DC/DC board to observe the impact of grounding. You should see significantly more voltage overshoot on the oscilloscope. Then connect ground cables again.

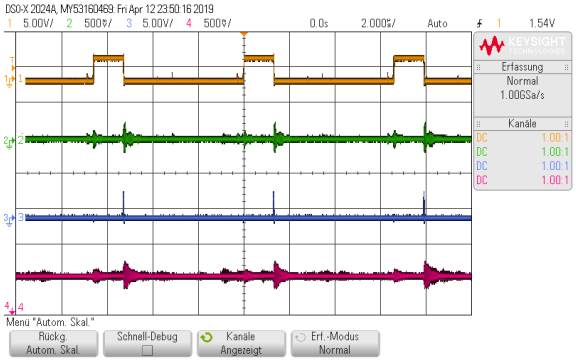
Push switch *sw*(1) = 1 corresponding to *LEN* = '1'. Output voltage should remain the same but PWM pulses become a little bit wider. Ok? Set back to *LEN* = '0' setting *sw*(1) = 0.

Set *sw*(0) = 1. Output voltage should remain the same but pulses become wider. And you can feel with your finger tip that the big blue resistor next to the green jumper will get warm after some seconds? Green line in Fig. 3.1 jumps to 1V. Reset load current to zero with *sw*(0)=0.

Saving the Scope's Screen to a File

- Save the screen on a USB memory stick. It must be FAT32 formatted.

(a) Oscillogam showing 4 channels



(b) Oscillogam showing 3 channels

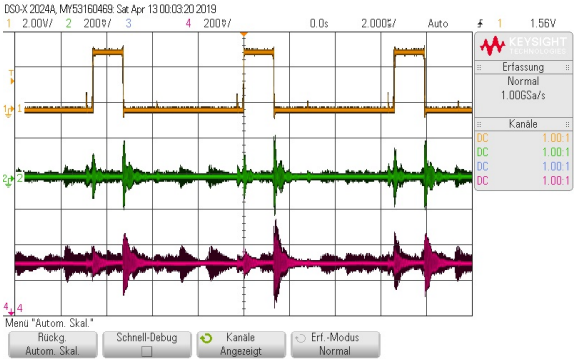


Fig. 3.1: Osci screen copies: CH1 (yellow): pulse-with modulated input signal. CH2 (green): voltage at 1Ω load resistor, CH3 (blue): synchronization signal, CH4 (red): output voltage.

3.2 The System Setup

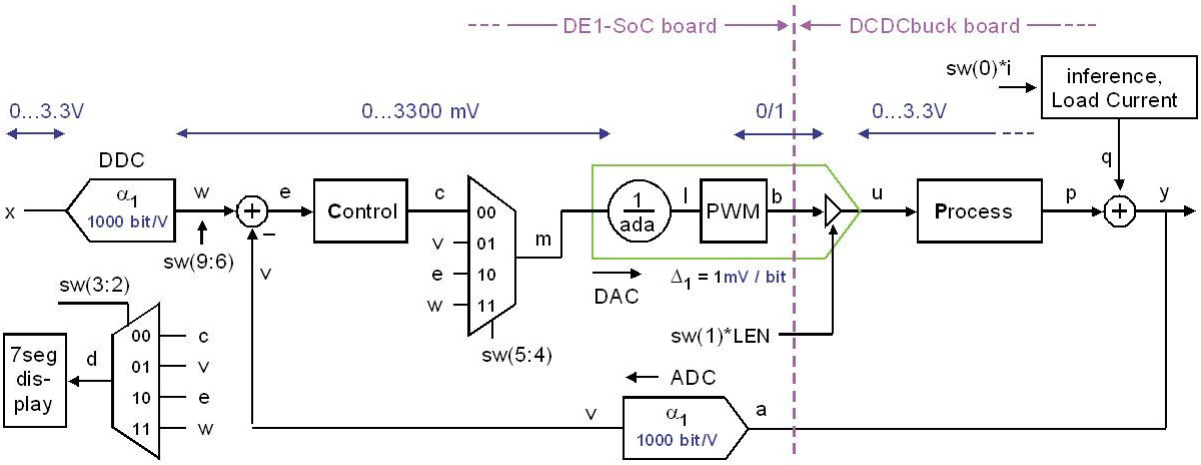


Fig. 3.2: The DC/DC buck converter schematic.

Fig 3.2 illustrates the system with its 2 number ranges:

1. Physical voltage range being mainly defined by the two output voltage levels U_{low}, U_{high} of the PWM DAC. Fig. 3.1 shows example voltages $U_{low} = 0.0V$, $U_{high} = 3.3V$ in blue letters.
2. ADC output range computed as $N_{ADCout} = \alpha_1 \cdot U_{ADCin} + \alpha_0$, with $\alpha_1 = 1000bits/V$, $\alpha_0 = 0V$ given by ADC *LTC2308* with output range 0…4095 corresponding to 0…4.095V. Consequently, *v, w* and *e* are given in millivolts (mV) and voltage range 0…3.3V corresponding to integral number range 0…3300. (To compute input *w* from *x* the amplification of the ADC in the feedback branch must be compensated for by the preceding digital-to-digital converter (DDC) with same gain α_1 as the ADC.)

Factor *ada* is shown as a part of the PWM DAC in Fig. 3.2. It compensates for different amplification of ADC and DAC, so that a direct series of these modules delivers an amplification of 1.

In detail: The PWM DAC with input level *l* in Fig. 3.2 translates a series of 0 … *pwm_period* ones within a period of *pwm_period* bits into voltage range U_{low} - U_{high}. Consequently, its amplification is $\Delta_1 = (U_{high} - U_{low}) / pwm\_period$. In the example of Fig. 3.1 with $pwm\_period = 330$ this corresponds to a DAC input level range of $l = 0$ … *pwm_period* resulting in $l = 0…330$. Thus, one bit of DAC input level *l* corresponds to $\Delta_1 = 10mV$ of DAC output voltage *u*.

Seven-segment display, switch *sw*(3:2)
Signals *c*, *v*, *e* and *w* can be selected by switches sw(3:2) to be shown on the 7-segment display in millivolt. Controller output *c* must be multiplied with $ada = \alpha1 \cdot \Delta1 = 10$ to be displayed in mV.

System operation mode, switch *sw*(5:4)
Signals *c*, *v*, *e* and *w* can be selected by switches sw(5:4) to be fed to the PWM DAC. Signals *v*, *e* and *w* must be divided by $ada = \alpha_1 \cdot \Delta_1$ to deliver a DAC output voltage range of $U_{low}…U_{high}$, whereas $U_{low} = 0.0V$, $U_{high} = 3.3V$, $ada = 10$ in the blue example of Fig. 3.1.

User Headers (according to listing 3.2)

- *pwmo* is the pulse-width modulated output signal. Max. number of bits is *pwm_period* which is by default 330, so that 3.3V are set by 330 bits → resolution 10 mV/bit.
- *enc* is a flag having a width of T_{clock}=20ns used as sync signal for the oscilloscope.
- *LEN* (= Low-side switch ENable) dis-/enables the low-side switching field effect transistor (FET) when 0/1.
- *ILoad_ON* switches a load current of I_L = 1A on when *ILoad_ON* = 1.
- *gpio_0* <= *gpio_1*: As user header *JP2* (= *gpio*_1) is covered by the *DCDCbuck* board, this *VHDL* statement copies all signals from *JP2* to free *JP1*, where they can be observed.

Listing 3.2: VHDL code lines of module *de1soc_DCDbuck(rtl_de1soc_DCDCbuck)*

```
-- DCDCbuck Daughter Board
gpio_1(0)  <= pwmo;     -- PIN01 = PWM output
gpio_1(2)  <= enc;      -- PIN03 = PWM enable out = contoller enable
gpio_1(16) <= LEN;      -- PIN19 = LEN (=Low driving FET Enable)
gpio_1(18) <= ILoad_ON; -- PIN21 = LoadCurrent_ON
-- miscellaneous
gpio_0 <= gpio_1; -- this line delivers more measurement points on gpio_0
```

Table 3.2: Functionality of switches *sw*(9:0) and push buttons *key*(3:0)

9	8	7	6	5	4	3	2	1	0
Set point *w*=1000·*x* in mV				DAC inp sel		7seg in sel		LEN	i_{Load}

Switches	
sw(9:6)	Select set point *x* in V
0000	Set point w = 0
0001	Set point w = 250
0010	Set point w = 500
0011	Set point w = 750
0100	Set point w = 1000
0101	Set point w = 1250
0110	Set point w = 1500
0111	Set point w = 1750
1000	Set point w = 2000
1001	Set point w = 2250
1010	Set point w = 2500
1011	Set point w = 2750
1100	Set point w = 3000
1101	Set point w = 3250
1110	Set point w = 3500
1111	Set point w = 3750
sw(5:4)	Select quantity fed to the input of the PWM DAC
00	control mode => output c
01	control mode => output v
10	control mode => output e
11	control mode => output w
sw(3:2)	Select quantity displayed on 7-segment display
00	display c in mV
01	display v in mV
10	display e in mV
11	display w in mV
sw(1)	Low-side driver ENable
0	Asynchronous mode: Low-side power-MOSFET is always off.
1	Synchronous mode: Low-side power-MOSFET operates invers to high-side driver
sw(0)	Load current switch
0	Load current OFF
1	Load current of 1A ON

Keys	(=push buttons)
key(0)	Global asynchronous reset, dominant over all other signals: all flipflops are reset to their reset-states
key(1)	Global enable: flipflops do no more change state when *key*(1) is pushed
key(2)	Load current ON: pushing *key*(2) has the same effect as *sw*(0)='1', current flow stops when *key*(2) is released.
key(3)	hold 7-segment display: 7seg-display is frozen while *key*(3) is pushed

3.3 Observing *DCDCbuck* Board Operation Details

3.3.1 *DCDCbuck* Board: Process and Load current Inference Realization

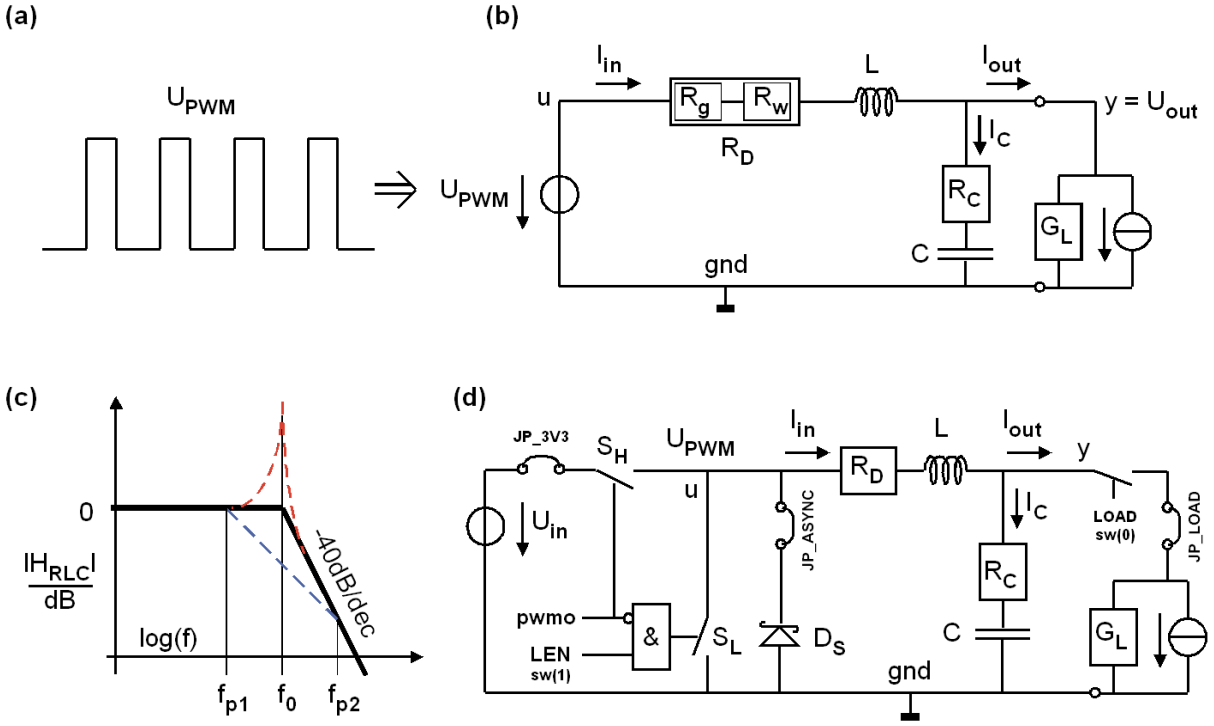


Fig. 3.3: Detailed view of process and load current inference of Fig. 3.2: **(a)** pulse-width modulated input voltage (U_{PWM}), **(b)** smoothed by an LC lowpass damped by some resistors, delivering **(c)** a transfer function over frequency f, and **(d)** is a more detailed view of part (b).

Fig. 3.3 details the boxes illustrated as *process* and *inference load current* in Fig. 3.2:

(a) Shows a pulse-width modulated input voltage (U_{PWM}) which is smoothed by an

(b) LC lowpass with unavoidable resistors R_D consisting of generator impedance Rg and wire resistance Rw, an equivalent serial resistor (R_C) of the capacitor and load conductance G_L. These resistors effect the damping of the LRLC lowpass, that is illustrating in the

(c) Bode diagram, showing asymptotes (**bold solid line**) over frequency f, with oscillating case indicated as red dashed line and aperiodic case shown as blue dashed line.

(d) Is a more detailed view of Fig. part (b), showing jumpers *JP_3V3*, *JP_ASYNC* and *JP_LOAD* being set. Consequently U_{PWM} oscillates between 0V and 3.3V, Schottky diode D_S is ready for supporting asynchronous operation enforced $sw(0)$ setting $LEN = 0$, and a *load current inference* of 1A can be imposed setting the *LOAD* switch $sw(1) = 1$.

High-side switch S_H and low-side switch S_L are controlled by the digital logic signal *pwmo* generated by the *FPGA*. A particular chip named *LM27222* [20] makes sure that S_H and S_L are operated non-overlapping, i.e. they newer conduct at the same time, because this would open a direct current path from U_{in} to ground.

3.3.2 Operating the *DCDCbuck* Board

Set the set point for the output voltage to $w = 1500\text{mV}$ and observe pulse with of *pwmo*, output voltage U_{out} and voltage over the 1Ω load resistor to know the current.

3.3.2.1 Uncontrolled Operation

We will now use fixed pulse-width for *pwmo* without control:

- Set *control mode => output w*.
- Load current off: $sw(0) = LOAD \rightarrow 0$.
- Set synchronous mode: $sw(1) = LEN \rightarrow 1$.
- Show $w = 1500\text{mV}$ on the 7-segment display, then switch to show *v*, which is measured by the ADC (*LTC2308*).

If we have pulses of 0/3.3V, the output voltage of the buck converter should be close to 1500mV.

Switch the load current on. The pulse-width should stay the same and the output voltage should drop by ΔU_{out}, typically some 200mV. This is helpful to compute R_D in Fig. 3.3(d) according to $R_D = \Delta U_{out}/I_{Load}$. What do you measure?

With load current ON, switch to the asynchronous mode. Switch S_L stays always open now. There should not be much change in the output voltage, except that we lose some more energy at the Schottky diode D_S, which is visible as a further output voltage drop.

Now switch the load current OFF. The output voltage begins to rise uncontrolled. This is because there is no more means available to pull U_{out} down: S_L is out of operation, no output current and Diode D_S is reverse biased.

3.3.2.2 Check Controlled Operation Roughly

We will now use adaptively controlled pulse-width for *pwmo*:

- Set *control mode => output c*.
- Load current off: $sw(0) = LOAD \rightarrow 0$.
- Set synchronous mode: $sw(1) = LEN \rightarrow 1$.
- Show $w = 1500\text{mV}$ on the 7-segment display, then switch to show *v*, which should be close to 1500.

Check output voltage roughly with and without load current, synchronous and asynchronous operation. The controller should now adapts the pulse-width on signal *pwmo* such, that the output voltage keeps constantly at some 1500mV.

3.3.2.3 Controlled Operation in Continuous Mode

For continuous mode, switch load current ON. Current through the inductor and low-side switch S_L is always in the same direction.

We now have the situation that output voltage is controlled by adapting pulse width. Less pulse-width at same output voltage and current corresponds to more efficiency.

Switch between synchronous and asynchronous mode: Enable and disable low-side switch S_L. using $sw(1) = LEN$. We see, that the converter operates more efficient (pulses are narrower) with LEN=1 as voltage drop across conduction switch S_L is less than through diode D_S.

Remove/set jumper *JP_ASYNC* several times to enable/disable diode D_S. We find that efficiency without D_S is lower but the circuit still works well, because low-side switch S_L has a built-in silicon diode. Schottky diode D_S has some 0.2V less forward voltage drop being more efficient and preventing switch S_L from heating.

3.3.2.4 Controlled Operation in Discontinuous Mode

For discontinuous mode, switch load current OFF. Current through the inductor and low-side switch S_L will now oscillate back and forth.

We still have the situation that output voltage is controlled by adapting pulse width and that less pulse-width at same output voltage and current corresponds to more efficiency.

Switch between synchronous and asynchronous mode: Enable and disable low-side switch S_L. using $sw(1) = LEN$. We see, that the converter operates more efficient (pulses are narrower) with LEN=0. This is because the operating low-side switch S_L (in sync mode) pulls inductor current into invers direction and leads it to ground. This is prevented in asynchronous mode by the diode. In the uncontrolled mode output voltage would rise uncontrolled, however, now in controlled mode pulse-with at signal pwmo is reduced to keep output voltage on a correct value.

Consequently, in cases of discontinuous operation due to low load currents the circuit operates more efficiently in the asynchronous mode.

3.4 Getting Along with the Digital Part

3.4.1 Load Current Sink

Read this carefully to not destroy the *DCDCbuck* board!

Avoid long phases of load current flow, particularly with high output voltages!

Setting *sw*(0)=1 or pressing push button *key*(2) turns the load current on, when jumper *JP_Load* (green in Fig. 3.0) is set to allow for load current flow.

Experiment: Set output voltage (labeled *y* in Fig. 3.2) to U_{out} = 1.2V ⇔ *sw*(9:6)="1000". Turn load current on ⇔ *sw*(0)='1' (or *key*(2) pushed) and "measure" with your finger tip how load resistor R_L (labeled *R_POWER* on the *DCDCbuck* board) becomes warm.

Load resistor R_L has to translate a maximum of $P_L = I_L \cdot U_L$=1A·1V = 1W of electrical power to thermal heat. This will mostly be done by transferring heat to surrounding air molecules.

The significantly smaller power FET (the "big" black square right to *R_POWER*) has to translate the rest of the load current's energy to thermal heat, that is $P_{FET} = I_L \cdot (Uout - 1V)$. This is performed be by transferring heat to surrounding air molecules and to the board. Take care! We have already "killed" several of those power FETs!

For an output voltage of x V, we have a total load power of 1A·xV = x W. The load resistor R_L translates 1W to thermal heat. The(smaller) power FET has to dissipate the remaining (x-1) W. If you tip your finger on the power FET to feel how it becomes hot, please do not touch its metal contacts as remaining salt of your fingers will cause leakage currents.

Exercise: Let U_{out} (= *y*) = 2.7V and load current on: I_L = 1A. What is the total load power that must be translated to heat, and witch power shares are dissipated by R_L and the power FET?

Total load power $P_{L,tot}$ =
. . . . . .

Resistor's power dissipation $P(R_L)$ =
. . . . . .

FET's power dissipation P(FET) =
. . . . . .

Functionality of the load current circuitry (Electronics students only)
Look at *Current Sink* schematics in Fig. 2.3 or download Eagle files *DCDCbuck_Rev5.sch* and *~.brd* from the author. Both *sw*(0) and pushing *key*(2) will drive the wire labeled *V_CURRENT* from logic '0' (0V) to logic '1' (3.3V). The non-inverting input voltage (pin 3) of operational amplifier (OA) *LM7301* gets 1/3 of this voltage, so that the pin 3 toggles between 0 and 1V. OA *LM7301will* do anything it can to get its inverting input (its pin *4*) to the same voltage as pin *3*. This can be achieved when $U_{out} \geq$ 1.2V. As the voltage at pin 4 is the voltage at the load resistor, R_L=1Ω, the load current is controlled to I_L = 1V/1Ω = 1A.

Solutions to the exercise above:

Total load power	$P_{L,tot}$	$= I_L \cdot U_{out} = 1A \cdot 2.7V = 2.7W$
Resistor's power dissipation	$P(R_L)$	$= I_L \cdot U(R_L) = 1A \cdot 1V = 1W$
FET's power dissipation	$P(FET)$	$= I_L \cdot U(FET) = 1A \cdot (U_{out}-1V) = 1.7W$

3.4.2 Testing the PWM DAC

With jumper settings as explained in chapter 3.0 and shown in Fig. 3.0, do the following switch settings (whereas "<=" is a VHDL signal driver):

- Set switches to: *w*=1200 [mV], pulse width: *l* = *w/ada*, *sw*(9:0) = 7-weg display shows *w, LEN*=0, Load current OFF . . . . . (solution see next page).
- In this mode the pulse-width is constant, whatever happens to the output voltage.
- Set point: *w* = 1200, corresponding to 1200mV when *DCDCbuck's* V_{CC}=3.3V.
- PWM DAC gets level *l* <= *w/ada* to translate set point *w* into mV of output voltage.
- 7-segement display gets *d* <= *w* to display set point (input) "`i 1200`".
- Synchronous mode on ⇔ LEN='1' (Low-side switch ENabled)
- Load current off.

You should now see the set point *w* = 1200 on the 7-segment display as string "`i 1200`".

- Switch displayed voltage to *l* <= *v*, displaying output voltage *y* as measured by the ADC.
- Displayed voltage should be around 1200 [mV].
- Compare *v* to output voltage *y* measured by CH4 of the oscilloscope. Should be similar.

Synchronous Mode Off ⇔ *LEN*<=0.
Output voltage *y* will now float up, because positive PWM pulses pump charges on the load capacitor while there is no way to discharge them again, as Low-side switch Enable (LEN) is 0 and the pull-down FET doesn't operate any more.

For electronics students: Current flowing out of the inductor's right side must flow in on its left side, also when the high-side switch is open. This is why we need the diode enabled with jumper labeled *Async* on the *DCDCbuck* board (pink jumper in Fig. 3.0). The Schottky diode is efficient but dispensable, as the deactivated power FET has an unavoidable build-in diode. However, it is not very efficient due to a high flow voltage, as the power FET are not optimized to operate as diodes. The difference may be 200mV flow voltage, not much but important to obtain high DC/DC conversion efficiency in asynchronous operation.

Load Current On
Within asynchronous mode of operation switch Load current on. Now the capacitor can be discharged over the load current and the output voltage falls some 100mV under the set point.

3.4.3 Testing Closed Loop DC/DC Buck Conversion Mode

Set switches to feed controller output *c* to the PWM DAC input *l* while *DCDCbuck* output voltage is fed back to the ADC, channel 0 (lower left pin of header *ADC CON*), 7-seg showing *w*, *LEN*=1, load current off:

sw(9:0) = ... (solution at the end of this page)

Now the pulse width should be automatically adopted to hold the output voltage constant, whatever load condition we have. you should see *w* indicated as "`i 1200`" on the 7-segment display.

Toggle *sw*(3) to *sw*(3)=0. Now the 7-seg displays should show process variable *v* which is the *DCDCbuck's* output voltage in mV as measured by the ADC.

Check this for different settings of set point *w*, synchronous and asynchronous mode, with and without load current. Does the system hold the output voltage constant? Should!

3.4.4 Testing the ADC

Set the output of your wave generator (oscilloscope output *WaveGen*) to DC, 2V and connect it (i) to a free channel on your oscilloscope and (ii) to the ADC input channel 0 (lower left pin of header *ADC CON*). Further set *sw*(3:2) to *sw*(3:2)="01", so that the 7-segment display shows the voltage of the *WaveGen*. Change output voltage of Wavegen in the range of 0…5V and compare the voltage measured with the oscilloscope and with the ADC. They should be very similar! Measure DC voltage of you oscilloscope with the **meas**urement menu.

Solution to 3.3.2 initial setting: *sw*(9:0) = `1000 11 11 00`
Solution to 3.3.3 initial setting: *sw*(9:0) = `1000 00 11 10`

3.5 Characterizing the RLC Lowpass (DC/DC Students only!)

What is your board number?

. . . . . .

3.5.1 Measuring Load Capacitors $C_{L\#}$

Measure the load capacitances on the *DCDCbuck* board. In the Electronics Laboratory of OTH Regensburg use *HM8118*.

Table 3.4.1: Load capacitance combinations

Device(s)	Capacitance nominal	Capacitance measured	Equivalent series resistor measured
C_{L1}	100 µF		
C_{L2}	330 µF		
C_{L3}	680 µF		
$C_{L1} + C_{L2}$	430 µF		
$C_{L1} + C_{L3}$	780 µF		
$C_{L2} + C_{L3}$	1010 µF		
$C_{L1} + C_{L2} + C_{L3}$	1110 µF		

3.5.2 Measuring Inductors $L_{\#}$

Measure inductance combinations on the *DCDCbuck* board. In the Electronics Laboratory of OTH Regensburg use *HM8118*.

Table 3.4.2: Load inductance combinations

Device(s)	Inductance nominal	Inductance measured	Equivalent series resistor measured
L_1	10 µH		
L_2	33 µH		
$L_1 + L_2$	43 µH		

3.5.3 Measuring Total Series Resistor R_D

Use switches to select constant pulse with and set points according to table 3.4.3. Measure average output voltage U_{out} (=*y* in Fig. 3.2), e.g. with oscilloscope or with a voltmeter.

Measure output voltage with and without load current and compute the total series resistor R_D from the resulting voltage difference: $R_D = (U_{out\,|\,IL=OFF} - U_{out\,|\,IL=ON}) / I_L$. Remove jumper *JP_Load* (green in Fig. 3.0) and replace it by an Ampere meter to get the accurate current.

Table 3.4.3: Total series resistor, measured from $R_D = (U_{out\,|\,IL=OFF} - U_{out\,|\,IL=ON}) / I_L$.

Set point *w*	$U_{out\,\mid\,IL=0}$	$U_{out\,\mid\,IL=ON}$	I_L *when ON*	R_D
1200				
1800				
2700				

3.5.4 Measuring a *Bode* Diagram of the RLC Low-Pass

3.5.4.1 Measure Point by Point

Open feedback loop wire between nodes *y* and *a* in Fig. 3.2 (orange wire in Figs. 2.1 and 3.0). Set switches to drive ADC output to PWM input, obtaining level *l <= v/ada*.

Connect DSO-X oscilloscope generator to CH 3 of the oscilloscope. (Other channels as described above: CH1: PWM output, CH2 V(R_{LOAD}), CH4 U_{out} (=*y*), probably Ext Trig IN to sync-signal pin3 of JP1.) Set WaveGen output to a DC-Signal of V_{CC} / 2 (should be 1.65V).

Feed the 1.65V from the oscilloscope's *WaveGen* output also to the ADC input channel 0 (node *a* in Fig. 3.2). You should observe that the U_{out} (node *y* in Fig. 3.2) at CH4 follows U_{in} at CH3. Modify WaveGen output in a range of $V_{CC}/2\pm1V$ to verify that.

With offset of V_{CC} / 2 (should be 1.65V) set *WaveGen* to add an AC amplitude at node *a* of $2V_{pp}$ (=2V peak-to-peak), i.e. a sinusoidal wave with swing $V_{CC}/2\pm1V$ at 10Hz. Measure output amplitude and phase of *y/a* and fill the gaps in table 3.4.4.

Table 3.4.4: Points of a *Bode* diagram

f / Hz =	10 Hz	50Hz	100	500	1000	5000	10000	
$\lvert y/a \rvert$ ($\lvert U_{out}/U_{in} \rvert$)								0.71
***angle*(*y/a*)**								45°

4 Conclusions

DCDCbuck board was operated as daughterboard of *DE1-SoC* board. Basic digital functionality was tested, *RLC* low pass was characterized. The reader is now ready for modeling and simulation with the parameters obtained.

5 References

[1] Available: https://www.terasic.com.tw.
[2] Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836
[3] Available: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=886
[4] Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836&PartNo=4
[5] CD ROM *DE1-SoC CD-ROM (rev.F Board) Version 5.1.2 of 2910-01-28* from [4]
[6] *DE1*-SoC User Manual, Ref. F, taken from [5], available: https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/DE1-SoC_UserManual.pdf
[7] *DE1*-SoC Schematic, Ref. F, taken from [5], available: https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/DE1-SoC_Schematic_revF.pdf
[8] K:\SB\Sources\EDA\Terasic\Hardware\
[9] Available: https://en.wikipedia.org/wiki/Intel_Quartus_Prime
[10] Available: https://en.wikipedia.org/wiki/ModelSim
[11] Av. https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html
[12] Available: https://edg.uchicago.edu/~tang/VHDLref.pdf
[13] Available: https://www.mimuw.edu.pl/~marpe/pul/card_vhdl.pdf
[14] Available: https://www.mimuw.edu.pl/~marpe/pul/card_1164.pdf
[15] Available: https://hps.hs-regensburg.de/scm39115/
[16] MAX4230 - MAX4324 data sheet, Maxm Integerated, Available: https://datasheets.maximintegrated.com/en/ds/MAX4230-MAX4234.pdf
[17] LT1711 / 1712 Rail-to-Rail Comparators, Analog Devices (Linear Technlogy) Available: https://www.analog.com/media/en/technical-documentation/data-sheets/171112f.pdf
[18] *Eagle*, *CadSoft Computer GmbH*, Available: http://ftp.cadsoft.de/freeware.htm.
[19] Kwang Liang Chong, "*Design, Manufacturing, Optimization and Characterization of a DC/DC Buck Converter Board Driving Up to 30V/4.5A for FPGA and µC Control*", BA Thesis, *OTH Regensburg, Electronics Lab*., Feb. 2019.
[20] Texas Instruments, *LM27222 High-Speed 4.5A Synchronous MOSFET Driver*, available: http://www.ti.com/lit/ds/symlink/lm27222.pdf