**EE 464**

**STATIC POWER CONVERSION-I**

**Spring 2022-2023**

**Homework 2**

**Complete Simulation Report**

Mehmet Emre Doğan – 2374825

Metehan Küçükler –

Table of Contents

[Introduction 2](#_Toc133778205)

[Magnetic Design 2](#_Toc133778206)

[Complete Simulations 2](#_Toc133778207)

[Conclusion 2](#_Toc133778208)

# Introduction

This report explains the design decisions for the hardware project. Furthermore, it presents the details of the Magnetic Design of the Isolated Power Supply and the simulation results for the selected topology.

# Magnetic Design

1. The duty range of the converter is selected as [0.278 – 0.336] to match the design by the Ti Webench. According to the duty range determination, the turn ratio is calculated via the MATLAB code below.

clearvars

syms d turnsRatio

v\_o = 48

d\_min = 0.278; v\_d\_minduty = 18;

d\_max = 0.366; v\_d\_maxduty = 12

turnsRatio\_minduty = ( (d\_min/(1-d\_min)) \* (v\_d\_minduty/v\_o) )^-1

turnsRatio\_maxduty = ( (d\_max/(1-d\_max)) \* (v\_d\_maxduty/v\_o) )^-1

According to the code above, the transformer turns ratio (Ns/Np) is calculated as 6.93.

2. The available cores and coil formers are investigated. Firstly, due to its available stock number is high, PCB5530-FA is selected as the coil former. Therefore, the compatible core 0P45530EC is selected as the transformer core. However, after calculations, it is seen that this core is overkill. Afterward, **79440A7 toroidal core is selected** due to its high stock number and wide window area. A wide window area makes the wounding procedure easier.
3. Using the MATLAB code below, the primary turn number is 13, while the secondary turn number is 87. The magnetizing inductance is 8 uH.

U\_o = v\_o;

v\_t = d\_max;

f\_sw = 100e3;

i\_out = 1;

i\_avgSec = i\_out/(1-v\_t);

xformerCurrRipple = 0.5; % percent

L\_sec = (U\_o\*(1-v\_t))/(xformerCurrRipple\*i\_avgSec\*f\_sw)

L\_pri = L\_sec/(turnsRatio\_maxduty^2)

% (turnsRatio\_maxduty^2)\*2.814e-6

syms priTurns secTurns

AL = 51e-9 % nH/T^2; minimal

priTurns = double(solve(L\_pri == AL\*priTurns^2))

secTurns = double(solve(L\_sec == AL\*secTurns^2))

% make sure core is not saturated

ampTurns = i\_out\*secTurns

1. According to the AWG table, the secondary should be wounded using 5 parallel 28 AWG wires. The primary, on the other hand, 18 parallel 28 AWG wires will be used.
2. According to the code below, the fill factor is 12.53%, which is reasonable.

windowArea\_mm2 = 427;

priTurns = ceil(priTurns(priTurns>0))

secTurns = ceil(secTurns(secTurns>0))

num\_of\_paralles\_pri = ceil(num\_of\_paralles\_pri)

num\_of\_paralles\_sec = ceil(num\_of\_paralles\_sec)

cableArea\_mm2 = 0.080;

primaryArea\_mm2 = priTurns\*num\_of\_paralles\_pri\*cableArea\_mm2

secondaryArea\_mm2 = secTurns\*num\_of\_paralles\_sec\*cableArea\_mm2

totalCableArea\_mm2 = primaryArea\_mm2 + secondaryArea\_mm2

fillFactor\_perc = 100\*totalCableArea\_mm2/windowArea\_mm2

1. Cable resistance calculation is done by the code below:

windingLengthPerTurn\_mm = 68.2

ohms\_per\_meter = 212.872 / 1e3

primaryLength\_m = windingLengthPerTurn\_mm \* priTurns \* 1e-3

secondaryLength\_m = windingLengthPerTurn\_mm \* secTurns \* 1e-3

primary\_DC\_resistance\_ohm = ohms\_per\_meter \* primaryLength\_m / num\_of\_paralles\_pri

secondary\_DC\_resistance\_ohm = ohms\_per\_meter \* secondaryLength\_m / num\_of\_paralles\_sec

The DC and AC resistances of the transformer are assumed equal thanks to the skin depth being greater than the radius.

**Primary Resistance: 10.5 mOhm**

**Secondary Resistance: 252 mOhm**

1. Copper losses are calculated by the code below:

diameter\_mm = vpa(0.32004\*u.mm)

radius\_mm = diameter\_mm/2

skinDepth\_cm = vpa(7.5/sqrt(f\_sw)\*u.cm)

skinDepth\_mm = unitConvert(skinDepth\_cm, u.mm)

% skin depth is greater than radius.

% Therefore, AC reistance equals DC resistance

DC\_to\_AC\_ratio = 1

primary\_AC\_resistance\_ohm = primary\_DC\_resistance\_ohm\*DC\_to\_AC\_ratio

secondary\_AC\_resistance\_ohm = secondary\_DC\_resistance\_ohm\*DC\_to\_AC\_ratio

resistancePri\_ohm = vpa(primary\_AC\_resistance\_ohm \* u.Ohm)

resistanceSec\_ohm = vpa(secondary\_AC\_resistance\_ohm \* u.Ohm)

copperLossPri = vpa(unitConvert((i\_in\_max\*u.A)^2 \* resistancePri\_ohm, u.W))

copperLossSec = vpa(unitConvert((i\_out\*u.A)^2 \* resistanceSec\_ohm, u.W))

copperLoss\_W = copperLossPri + copperLossSec

**Total Copper Losses: 0.42 W**

1. Core losses are calculated by the code below:

permeability = 26;

mu\_zero = 1.25663706212e-6;

pathLength\_m = 107e-3;

fluxDensity\_Tesla = mu\_zero \* permeability \* ampTurns / pathLength\_m

% using graph above, 0.03 Tesla @ 100 kHz corresponds to

wattLoss\_mW\_cm3 = 60\*u.mW/u.cm^3

volume\_mm3 = 21300;

volume\_cm3 = vpa(unitConvert(volume\_mm3\*u.mm^3, u.cm^3))

coreLoss\_w = vpa(unitConvert(wattLoss\_mW\_cm3 \* volume\_cm3, u.W))

**Core Loss: 1.27 W**

The core loss is comparable with the copper loss. Hence, the design is good. No need to iterate more.

# Complete Simulations

# Conclusion

# Appendix – 1

calc.mlx

clearvars

u = symunit;

syms d turnsRatio

format shortEng

% format short

v\_o = 48

d\_min = 0.278; v\_d\_minduty = 18;

d\_max = 0.366; v\_d\_maxduty = 12

turnsRatio\_minduty = ( (d\_min/(1-d\_min)) \* (v\_d\_minduty/v\_o) )^-1

turnsRatio\_maxduty = ( (d\_max/(1-d\_max)) \* (v\_d\_maxduty/v\_o) )^-1

U\_o = v\_o;

v\_t = d\_max;

f\_sw = 100e3;

i\_out = 1;

i\_avgSec = i\_out/(1-v\_t);

xformerCurrRipple = 0.5; % percent

L\_sec = (U\_o\*(1-v\_t))/(xformerCurrRipple\*i\_avgSec\*f\_sw)

L\_pri = L\_sec/(turnsRatio\_maxduty^2)

% (turnsRatio\_maxduty^2)\*2.814e-6

syms priTurns secTurns

AL = 51e-9 % nH/T^2; minimal

priTurns = double(solve(L\_pri == AL\*priTurns^2))

secTurns = double(solve(L\_sec == AL\*secTurns^2))

% make sure core is not saturated

ampTurns = i\_out\*secTurns

**AWG selection**

p\_o = i\_out \* v\_o

i\_in\_max = v\_o/v\_d\_maxduty

selectedAWGRating = 0.226;

num\_of\_paralles\_sec = i\_out/selectedAWGRating

num\_of\_paralles\_pri = i\_in\_max/selectedAWGRating

**Fill Factor Calculation**

windowArea\_mm2 = 427;

priTurns = ceil(priTurns(priTurns>0))

secTurns = ceil(secTurns(secTurns>0))

num\_of\_paralles\_pri = ceil(num\_of\_paralles\_pri)

num\_of\_paralles\_sec = ceil(num\_of\_paralles\_sec)

cableArea\_mm2 = 0.080;

primaryArea\_mm2 = priTurns\*num\_of\_paralles\_pri\*cableArea\_mm2

secondaryArea\_mm2 = secTurns\*num\_of\_paralles\_sec\*cableArea\_mm2

totalCableArea\_mm2 = primaryArea\_mm2 + secondaryArea\_mm2

fillFactor\_perc = 100\*totalCableArea\_mm2/windowArea\_mm2

**Cable Resistance Calculation**

windingLengthPerTurn\_mm = 68.2

ohms\_per\_meter = 212.872 / 1e3

primaryLength\_m = windingLengthPerTurn\_mm \* priTurns \* 1e-3

secondaryLength\_m = windingLengthPerTurn\_mm \* secTurns \* 1e-3

primary\_DC\_resistance\_ohm = ohms\_per\_meter \* primaryLength\_m / num\_of\_paralles\_pri

secondary\_DC\_resistance\_ohm = ohms\_per\_meter \* secondaryLength\_m / num\_of\_paralles\_sec

**Copper Loss Calculation**

diameter\_mm = vpa(0.32004\*u.mm)

radius\_mm = diameter\_mm/2

skinDepth\_cm = vpa(7.5/sqrt(f\_sw)\*u.cm)

skinDepth\_mm = unitConvert(skinDepth\_cm, u.mm)

% skin depth is greater than radius.

% Therefore, AC reistance equals DC resistance

DC\_to\_AC\_ratio = 1

primary\_AC\_resistance\_ohm = primary\_DC\_resistance\_ohm\*DC\_to\_AC\_ratio

secondary\_AC\_resistance\_ohm = secondary\_DC\_resistance\_ohm\*DC\_to\_AC\_ratio

resistancePri\_ohm = vpa(primary\_AC\_resistance\_ohm \* u.Ohm)

resistanceSec\_ohm = vpa(secondary\_AC\_resistance\_ohm \* u.Ohm)

copperLossPri = vpa(unitConvert((i\_in\_max\*u.A)^2 \* resistancePri\_ohm, u.W))

copperLossSec = vpa(unitConvert((i\_out\*u.A)^2 \* resistanceSec\_ohm, u.W))

copperLoss\_W = copperLossPri + copperLossSec

**Core Loss Calculation**

çizelge içeren bir resim

Açıklama otomatik olarak oluşturuldu

permeability = 26;

mu\_zero = 1.25663706212e-6;

pathLength\_m = 107e-3;

fluxDensity\_Tesla = mu\_zero \* permeability \* ampTurns / pathLength\_m

% using graph above, 0.03 Tesla @ 100 kHz corresponds to

wattLoss\_mW\_cm3 = 60\*u.mW/u.cm^3

volume\_mm3 = 21300;

volume\_cm3 = vpa(unitConvert(volume\_mm3\*u.mm^3, u.cm^3))

coreLoss\_w = vpa(unitConvert(wattLoss\_mW\_cm3 \* volume\_cm3, u.W))