1KW 4 SWITCH BUCK BOOST CONVERTER DESIGN WITH LT8705A

Ratings

```
Power=1000;
Vout=28;
Iout=1000/28;
VinMin=18;
VinMax=36;
RLoad=Vout/Iout;
```

Operating Conditions

```
D3maxBoost=1-VinMin/Vout;
D2minBuck=1-Vout/VinMax;
Roscillation=436.15;                                    %in kohm
frequency=43715000/(Roscillation+1);
tminBuck=260*10^-9;

IInductorBuck=Iout;
IInductorBoost=Power/VinMin;
IInductorRippleBoost=IInductorBoost*0.3;

%Following minimum inductance values are calculated based on the formulas
%in the datasheet
L=VinMin*D3maxBoost/IInductorRippleBoost/frequency;

IInductorRippleBuck=tminBuck*frequency*Vout/frequency/L;
IInductorPeakBoost=IInductorBoost+IInductorRippleBoost*0.5;
IInductorPeakBuck=IInductorBuck+IInductorRippleBuck*0.5;
```

Sense Resistor Calculation

```
%Values and formulas are provided in the datasheet
VRsenseBoost=0.107;
VRSenseBuck=0.086;
RSenseBoost=2*VRsenseBoost*VinMin/((2*Power)+(IInductorRippleBoost*VinMin));
RSenseBuck=2*VRSenseBuck/(2*Iout-IInductorRippleBuck);
RSenseRecommended=min(RSenseBoost,RSenseBuck)/1.3;
Lmin1Boost=VinMin*D3maxBoost/(2*frequency*(VRsenseBoost/RSenseRecommended-Power/VinMax));
Lmin2Boost=(Vout-VinMin*Vout/(Vout-VinMin))*RSenseRecommended/0.08/frequency;
Lmin1Buck=VinMax*(1-Vout/(VinMax-Vout))*RSenseRecommended/0.08/frequency;
```

Output Ripple Calculations

```
[numberOfCaps, VoutRipple, totalNumberOfCaps, desiredOperationRipple]=CalculateOutputRipple(150
```

MOSFET Power Loss Calculations for Given MOSFETHe

```
[requiredTJA, LossM1, LossM3, LossM4]=CalculateBoostModeSwitchPowerDissipation(6, 4.5, 5.5, 70,
```

Displaying Values

```
disp('INDUCTOR & INDUCTOR CURRENT & DUTY CYCLE');
```

```
INDUCTOR & INDUCTOR CURRENT & DUTY CYCLE
```

```
disp(['Maximum Duty Cycle of third Mosfet: %',num2str(D3maxBoost*100)]);
```

```
Maximum Duty Cycle of third Mosfet: %35.7143
```

```
disp(['Inductor Current Value in Boost Region: ',num2str(IInductorBoost),'A']);
```

```
Inductor Current Value in Boost Region: 55.5556A
```

```
disp(['Inductor Current Value in Buck Region: ',num2str(IInductorBuck),'A']);
```

```
Inductor Current Value in Buck Region: 35.7143A
```

```
disp(['Inductor Ripple Current Value in Boost Region: ', num2str(IInductorRippleBoost),'A']);
```

```
Inductor Ripple Current Value in Boost Region: 16.6667A
```

```
disp(['Inductor Ripple Current Value in Buck Region: ',num2str(IInductorRippleBuck),'A']);
```

```
Inductor Ripple Current Value in Buck Region: 1.8874A
```

```
disp(['Inductor Current Peak in Buck Region: ',num2str(IInductorPeakBuck),'A']);
```

```
Inductor Current Peak in Buck Region: 36.658A
```

```
disp(['Inductor Current Peak in Boost Region: ', num2str(IInductorPeakBoost),'A']);
```

```
Inductor Current Peak in Boost Region: 63.8889A
```

```
disp(['Minimum Inuctor Current Satisfying the Ripple Current Condition: ',num2str(L*10^6),' mic
```

```
Minimum Inuctor Current Satisfying the Ripple Current Condition: 3.8571 micro Henry
```

```
disp(' ');
```

```
disp('SATISFACTION OF INDUCTANCE CONDITIONS & SENSE RESISTOR');
```

```
SATISFACTION OF INDUCTANCE CONDITIONS & SENSE RESISTOR
```

```
if L>Lmin1Buck && L>Lmin1Boost && L>Lmin2Boost
    disp('Chosen Inductance Value Satisfies the Minimum Inductance Conditions')
else
    disp('Chosen Inductance Value Does Not Satisfy the Minimum Inductance Conditions')
end
```

```
Chosen Inductance Value Satisfies the Minimum Inductance Conditions
```

```
disp(['Calculated Inductor Current Sense Resistor: ',num2str(RSenseRecommended*1000),' mili Ohm
```

```
Calculated Inductor Current Sense Resistor: 1.2883 mili Ohm
```

```
disp(' ');
```

```
disp('OUTPUT CAPACITOR');
```

```
OUTPUT CAPACITOR
```

```
disp(['Number of Capacitors Needed If Specified Capacitor is Used: ', num2str(numberOfCaps)]);
```

```
Number of Capacitors Needed If Specified Capacitor is Used: 8
```

```
disp(['Output Ripple Calculated Based on Used Capacitors: ', num2str(VoutRipple), ' Volts']);
```

```
Output Ripple Calculated Based on Used Capacitors: 0.26257 Volts
```

```
if(totalNumberOfCaps-numberOfCaps)
    disp(['Additional Number of Capacitors Requested: ', num2str(totalNumberOfCaps-numberOfCaps
    disp(['Corresponding Ripple for Requested Number Of Caps Added: ', num2str(desiredOperation
end
```

```
Additional Number of Capacitors Requested: 2
Corresponding Ripple for Requested Number Of Caps Added: 0.21005 Volts
```

```
disp(' ');
```

```
disp('POWER LOSS ON SWITCHES & REQUIRED THERMAL RESISTANCE FOR LOSSSES');
```

```
POWER LOSS ON SWITCHES & REQUIRED THERMAL RESISTANCE FOR LOSSSES
```

```
disp(['Power disspiations on Switches respectively for given MOSFET; SW1: ', num2str(LossM1),
```

```
Power disspiations on Switches respectively for given MOSFET; SW1: 18.5185W  SW2: 0 (Ignoring start-up)W   SW3: 1.15
```

```
disp(['Required Junction to Ambient thermal resistance for safe operation: R_thJA= ',num2str(re
```

```
Required Junction to Ambient thermal resistance for safe operation: R_thJA= 3.51C/W
```

Appendix-2: Switch Power Dissipation Calculator for Given MOSFET Ratings

```
function [requiredTJA,M1Power, M3Power, M4Power]=CalculateBoostModeSwitchPowerDissipation(RDSOn
   RDSOn=RDSOnInMiliOhm*0.001;
   tr=trInNanoSec*10^-9;
   tf=tfInNanoSec*10^-9;
   M1Duty=1;
```

```
    M3Duty=1-Vin/Vout;
    M4Duty=1-M3Duty;
    ILoad=Power/Vout;
    IIn=ILoad/M4Duty;

    %M1 loss
    IM1rms=M1Duty*IIn;
    conductionLossM1=RDSOn*IM1rms^2;

    %M3 loss
    IM3rms=M3Duty*ILoad;
    conductionLossM3=RDSOn*IM3rms^2;
    switchingLossM3=0.5*Vout*(tr+tf)*freq*IM3rms;

    %M4 loss
    IM4rms=M4Duty*ILoad;
    conductionLossM4=RDSOn*IM4rms^2;
    switchingLossM4=0.5*Vout*(tr+tf)*freq*IM4rms;

    M1Power=conductionLossM1;
    M3Power=conductionLossM3+switchingLossM3;
    M4Power=conductionLossM4+switchingLossM4;
    TJA=[(Tmax-Tamb)/M1Power,(Tmax-Tamb)/M3Power,(Tmax-Tamb)/M4Power];
    requiredTJA=min(TJA);
end
```

Appendix-3: Switch Power Dissip Calculator for Given MOSFET Ratings

```
%             will not be used this script as there will be used single MOSFET per gate
%             power will be handled by heatsinks and layout

% function [numberOfM1, M1Power, numberOfM3, M3Power, numberOfM4, M4Power]=CalculateBoostModeNu
%    RDSOn=RDSOnInMiliOhm*0.001;
%    tr=trInNanoSec*10^-9;
%    tf=tfInNanoSec*10^-9;
%    M1Duty=1;
%    M3Duty=1-Vin/Vout;
%    M4Duty=1-M3Duty;
%    conductionLossM1=10;
%    conductionLossM3=10;
%    conductionLossM4=10;
%    switchingLossM3=10;
%    switchingLossM4=10;
%    ILoad=Power/Vout;
%    IIn=ILoad/M4Duty;
%
%    %M1 loss
%    n=1;
%     while Tamb+conductionLossM1*RthermalJA>Tmax
%         IM1rms=M1Duty*IIn/n;
```

```
%         conductionLossM1=RDSOn*IM1rms^2;
%         if(n>3)
%             disp('Too many MOSFETs in parallel are needed!!');
%             disp('Do not use this MOSFET as SW1!!!!');
%             break;
%         end
%         n=n+1;
%     end
%     numberOfM1=n;
%     M1Power=conductionLossM1;
%
%     %M3 loss
%     n=1;
%     while Tamb+(conductionLossM3+switchingLossM3)*RthermalJA>Tmax
%         IM3rms=(M3Duty*ILoad/n);
%         conductionLossM3=RDSOn*IM3rms^2;
%         switchingLossM3=0.5*Vout*(tr+tf)*freq*IM3rms;
%         if(n>3)
%             disp('Too many MOSFETs in parallel are needed!!');
%             disp('Do not use this MOSFET as SW3!!!!');
%             break;
%         end
%         n=n+1;
%     end
%     numberOfM3=n;
%     M3Power=conductionLossM3+switchingLossM3;
%
%     %M4 loss
%     n=1;
%     while Tamb+(conductionLossM4+switchingLossM4)*RthermalJA>Tmax
%         IM4rms=(M4Duty*ILoad/n);
%         conductionLossM4=Vsd*IM4rms;
%         switchingLossM4=0.5*Vout*(tr+tf)*freq*IM4rms;
%         if(n>3)
%             disp('Too many MOSFETs in parallel are needed!!');
%             disp('Do not use this MOSFET as SW4!!!!');
%             break;
%         end
%         n=n+1;
%     end
%     numberOfM4=n;
%     M4Power=conductionLossM4+switchingLossM4;
% end
```

Appendix-4: Output Ripple Calculator for Given Capacitor Ratings

```
function [requiredNumOfCaps, ripple,requiredNumOfCapsUpdated, rippleUpdated]=CalculateOutputRip
if~exist('numberOfAdditionalCaps','var')
    numberOfAdditionalCaps=0;
end
```

```
VinMin=18;
Vout=28;
n=1;
Iout=Power/Vout;
var1=1;
    while var1>0.01*Vout
        n=n+1;
        %%Output Ripple Formula provided in the datasheet
        var1=Iout*(Vout-VinMin)/freq/(n*capacitanceInuF*10^-6)/VinMin+Power*(ESRInmOhm*10^-3/n)
        if(n>10)
            disp('Too many capacitor is needed to satisfy ripple conditions.');
            disp(' Do not use this capacitor for your application!');
            break;
        end
    end
    if(mod(n,2))
        requiredNumOfCaps=n+1;
        ripple=Iout*(Vout-VinMin)/freq/((n+1)*capacitanceInuF*10^-6)/VinMin+Power*(ESRInmOhm*10
        disp(['Required number of specified capacitor is odd.', 'For layout it is increased by
    else
        ripple=var1;
        requiredNumOfCaps=n;
    end
    if(numberOfAdditionalCaps)
        requiredNumOfCapsUpdated=numberOfAdditionalCaps+requiredNumOfCaps;
        rippleUpdated=Iout*(Vout-VinMin)/freq/(requiredNumOfCapsUpdated*capacitanceInuF*10^-6)/
    else
        requiredNumOfCapsUpdated=requiredNumOfCaps;
        rippleUpdated=ripple;
    end
end
```