Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

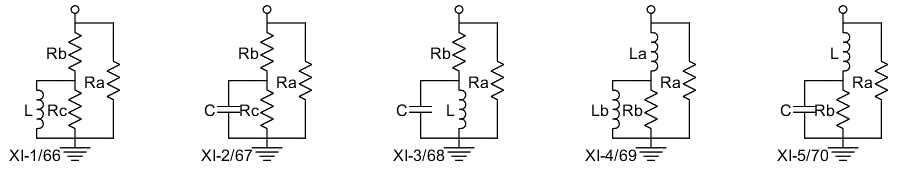
Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated



Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

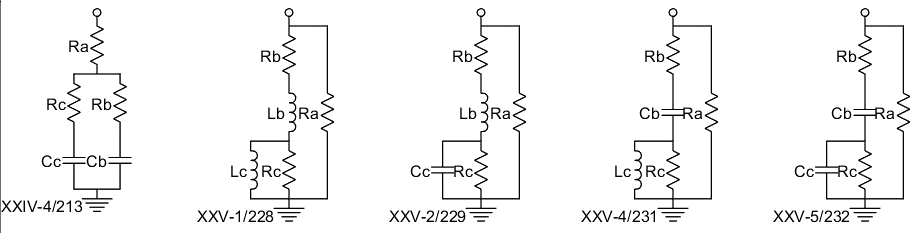
Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated



Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

MATLAB code that generates this census:

function theCompIdList = Census(theCat)

% CompList = CRCat.Census Do a census of components meeting the

% criteria coded below. CompIdList is an array of all the component

% Id's (i.e., indices in Cat.list) that are found in the census.

% jcr21Apr2023

% Build and print out a list of all components that have one or

% two L or C elements. Display their schematics. List

% equivalent components. jcr21Apr2023

% Find the comps that meet the criteria.

for aIndex = length(theCat.list) : -1 : 1

aComp = theCat.list(aIndex); % Component to be checked.

aVstr = string( aComp.symVn ); % Symbolic variable names for the component, convert to string.

aNumLC = sum( strncmp(aVstr,"L",1) ) + sum( strncmp(aVstr,"C",1) );

aCompLogical(aIndex) = ( aNumLC == 1 || aNumLC == 2 ); % This is the criteria.

end % for aIndex

% Get the Id's, i.e., indices of the qualifying components.

theCompIdList = find( aCompLogical );

% List out the name of each component, its variable list, and

% equivalent components.

for aCompId = theCompIdList

fprintf( '%11s ',theCat.Id2Name(aCompId) );

% fprintf('%s',string( theCat.list(aCompId).symVn )); % This prints out each entire variable name.

aStr = string( theCat.list(aCompId).symVn );

for aIndex=1:length(aStr)

fprintf( '%c',aStr{aIndex}(1) ); % This prints out only the first letter of each variable name.

end % for aIndex

aConv = theCat.Same( aCompId );

if ~isempty( aConv.fewer )

for aId = aConv.fewer

if aId ~= aConv.fromId

fprintf( ' %s',theCat.Id2Name(aId) );

end % if aId

end % for aId

fprintf(' (Component is degenerate)');

elseif ~isempty( aConv.equal )

for aId = aConv.equal

if aId ~= aConv.fromId

fprintf( ' %s',theCat.Id2Name(aId) );

end % if aId

end % for aId

end % if length

fprintf('\n');

end % for aComp

% Draw component schematics, in groups of 10.

for aIndex = 1 : 10 : length(theCompIdList)

theCat.DrawArray( theCompIdList(aIndex:end) );

input('='); % Wait for user to input something before doing the next plot.

end % for aIndex

end % CRCat.Census