```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 18 18 594 774
%%Title: $RCSfile: pcb_stackup.eps,v $
%%Creator: Daniel C. Nygren
%%CreationDate: $Date: 2010/08/12 22:12:08 $
%%EndComments

% *** Enter your fab title here ***
/fab_name (Project_Name Board_Name PCB Stackup 271-XXXX-YY rev ZZ) def %Enter name of fab

% *** Enter your stackup unit names here ***
/cu_weight_units (oz) def          % Usually ounces (oz)
/layer_thickness_units (mils) def  % Usually thousands of an inch (mils)

% If you want to omit a layer variable, use a null string "()".
%   [(Layer Name)    (Layer#) (Cu Weight) (Thickness) (GrayScale) -> 1.0 White, 0.0 Black]
/layer_stackup [
    [(Top)             (1)     (0.5)       (0.59)       (0.9)]
    [(Prepreg 1x1080)  ()      ()          (4.02)       (1.0)]
    [(02_GND_01)       (2)     (2.0)       (2.36)       (0.6)]
    [(Core 2x1080)     ()      ()          (5.90)       (1.0)]
    [(03_PWR_01)       (3)     (2.0)       (1.18)       (0.4)]
    [(Prepreg 2x2313)  ()      ()          (6.46)       (1.0)]
    [(04_GND_02)       (4)     (2.0)       (2.36)       (0.6)]
    [(Core 2x1080)     ()      ()          (5.90)       (1.0)]
    [(05_PWR_02)       (5)     (2.0)       (2.36)       (0.4)]
    [(Prepreg 2x2313)  ()      ()          (6.46)       (1.0)]
    [(06_PWR_03)       (6)     (2.0)       (2.36)       (0.4)]
    [(Core 2x1080)     ()      ()          (5.90)       (1.0)]
    [(07_GND_03)       (7)     (2.0)       (2.36)       (0.6)]
    [(Prepreg 2x1080)  ()      ()          (3.90)       (1.0)]
    [(08_SIG_01)       (8)     (1.0)       (1.18)       (0.8)]
    [(Core 2x2116)     ()      ()          (9.84)       (1.0)]
    [(09_SIG_02)       (9)     (1.0)       (1.18)       (0.8)]
    [(Prepreg 2x1080)  ()      ()          (3.90)       (1.0)]
    [(10_GND_04)       (10)    (2.0)       (2.36)       (0.6)]
    [(Core 2x1080)     ()      ()          (5.90)       (1.0)]
    [(11_PWR_04)       (11)    (2.0)       (2.36)       (0.4)]
    [(Prepreg 2x2313)  ()      ()          (6.46)       (1.0)]
    [(12_PWR_05)       (12)    (2.0)       (2.36)       (0.4)]
    [(Core 2x1080)     ()      ()          (5.90)       (1.0)]
    [(13_GND_05)       (13)    (2.0)       (2.36)       (0.6)]
    [(Prepreg 2x2313)  ()      ()          (6.46)       (1.0)]
    [(14_PWR_06)       (14)    (2.0)       (2.36)       (0.4)]
    [(Core 2x1080)     ()      ()          (5.90)       (1.0)]
    [(15_GND_06)       (15)    (2.0)       (2.36)       (0.6)]
    [(Prepreg 1x1080)  ()      ()          (4.02)       (1.0)]
    [(Bottom)          (16)    (0.5)       (0.59)       (0.9)]
] def

% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
% ^^^^^^^^^^ You should only have to change things above here ^^^^^^^^^^
% ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

%% Below are some example stackups you can uncomment and use as a starting
%% point for new designs.  These stackups try to:
%% - Have signal layers adjacent to a power or ground plane.
%% - Have internal signal layers shielded between planes.
%% - Have power and ground planes adjacent.
%% - Have multiple ground planes when possible.

%% Here's an ten layer stackup w/o thickness data for reference.
%% If you want to omit a layer variable, use a null string "()".
%%   [(Layer Name)    (Layer#) (Cu Weight) (Thickness) (GrayScale) -> 1.0 White, 0.0 Black]
%/layer_stackup [
%    [(Top)             (1)     (0.5)       ()       (0.9)]
%    [(02_GND_01)       (2)     (1.0)       ()       (0.6)]
%    [(03_SIG_01)       (3)     (1.0)       ()       (0.8)]
%    [(04_SIG_02)       (4)     (1.0)       ()       (0.8)]
%    [(05_PWR_01)       (5)     (1.0)       ()       (0.4)]
```

```
%    [(06_GND_02)        (6)     (1.0)        ()        (0.6)]
%    [(07_SIG_03)        (7)     (1.0)        ()        (0.8)]
%    [(08_SIG_04)        (8)     (1.0)        ()        (0.8)]
%    [(09_GND_03)        (9)     (1.0)        ()        (0.6)]
%    [(Bottom)          (10)     (0.5)        ()        (0.9)]
%] def

% Here’s an eight layer stackup w/o thickness data for reference.
%% If you want to omit a layer variable, use a null string "()".
%%   [(Layer Name)    (Layer#) (Cu Weight) (Thickness) (GrayScale) -> 1.0 White, 0.0 Black]
%/layer_stackup [
%    [(Top)              (1)     (0.5)        ()        (0.9)]
%    [(02_PWR_01)        (2)     (1.0)        ()        (0.4)]
%    [(03_GND_01)        (3)     (1.0)        ()        (0.6)]
%    [(04_SIG_01)        (4)     (1.0)        ()        (0.8)]
%    [(05_SIG_02)        (5)     (1.0)        ()        (0.8)]
%    [(06_GND_02)        (6)     (1.0)        ()        (0.6)]
%    [(07_PWR_02)        (7)     (1.0)        ()        (0.4)]
%    [(Bottom)           (8)     (0.5)        ()        (0.9)]
%] def

% Here’s an six layer stackup w/o thickness data for reference.
%% If you want to omit a layer variable, use a null string "()".
%%   [(Layer Name)    (Layer#) (Cu Weight) (Thickness) (GrayScale) -> 1.0 White, 0.0 Black]
%/layer_stackup [
%    [(Top)              (1)     (0.5)        ()        (0.9)]
%    [(02_GND_01)        (2)     (1.0)        ()        (0.6)]
%    [(03_SIG_01)        (3)     (1.0)        ()        (0.8)]
%    [(04_SIG_02)        (4)     (1.0)        ()        (0.8)]
%    [(05_PWR_01)        (5)     (1.0)        ()        (0.4)]
%    [(Bottom)           (6)     (0.5)        ()        (0.9)]
%] def

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pcb_stackup.eps - Version $Revision: 1.2 $: A PCB display program in PostScript
% Release $Name:  $ (Only defined if checked out as a specific release)
%
% by Daniel C. Nygren $Date: 2010/08/12 22:12:08 $
% E-mail: Dan.Nygren@oracle.com
% Permanent E-mail: Dan.Nygren@alumni.clemson.edu
%
% Copyright 2010 by Oracle America, Confidential - Oracle Internal
%
%    I wrote this program because I wanted an automated graphical method of
% documenting PCB layer stackups. Users edit the layer_stackup array at the
% start of this file and view with a PostScript viewer, print on a PostScript
% printer, import this EPS file into another application (Staroffice etc.), or
% convert to another image format to view the stackup image.
%
%    The PostScript layer_stackup array that a user edits to create a stackup
% has been moved to the start of the file so users would not be intimidated by
% the PostScript code contained in the body of this program.  The array is set
% up to contain each entry as a string, even though some entries are not
% treated as such by the program.  This was done to simplify data entry. A user
% always enters data as a string and the program converts the data to the
% proper format as appropriate. The fab_name printed at the top of the page
% is scaled to fit the page no matter how long this title string is. The total
% thickness of the board is printed at the bottom of the page if the thickness
% is greater than zero. In many cases the user omits the thicknesses because
% the output is just used for discussion purposes and a final tally isn’t
% needed.
%
% CALLING SEQUENCE      N/A
%
% EXAMPLES  (View on screen using Gnome Ghostview)
%           /usr/bin/ggv pcb_stackup.eps
%           (View on screen using Ghostscript)
%           /pkg/gnu/bin/gs pcb_stackup.eps
%           (View on paper if using a PostScript Printer)
%           lp pcb_stackup.eps
```

```
%           (Convert to a GIF w/ transparent background using Imagemagick)
%           /pkg/local/bin/convert pcb_stackup.eps pcb_stackup.gif
%           (Convert to a PNG w/ white background using Imagemagick)
%           /pkg/local/bin/convert -flatten pcb_stackup.eps pcb_stackup.gif
%
%
% TARGET SYSTEM         PostScript Level 2 Interpreters
%
% DEVELOPMENT SYSTEM    Ghostscript http://www.ghostscript.com/
%                       Xerox WorkCentre 4250
%
% CALLS                 N/A
%
% CALLED BY             N/A
%
% INPUTS                fab_name, layer_array, cu_weight_units, layer_thickness_units
%
% OUTPUTS               A PostScript interpreter dirties up clean sheets of paper
%                       or the screen with the output of this program.
%
% RETURNS               N/A
%
% ERROR HANDLING        N/A
%
% WARNINGS      1) This program requires PostScript Level 2.
%               2) A PostScript dictionary for this program is not
%               created since most interpreters running this program
%               make a copy of usrdict for use by EPS programs.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               REVISIONS
%
% $Log: pcb_stackup.eps,v $
% Revision 1.2  2010/08/12 22:12:08  nygren
% Updated comments
%
% Revision 1.1  2010/08/05 16:53:40  nygren
% Initial revision
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% By convention, all my variables have a underscore in them, and my
% procedures start with a capital letter. (Except for the ubiquitous "inch")

% I use bind to avoid the slow name lookup on the commonly used inch procedure
/inch {72 mul} bind def %Define an inch as 72 points

% The bounding box at the start of this program was calculated by
% taking llx and lly as the margins, and urx and ury as the paper
% width and height minus the margin. Remember to change the bounding
% box if you change the paper parameters.
/paper_width {8.5} def  %Enter paper width in inches (portrait)
/paper_height {11} def  %Enter paper height in inches (portrait)
/paper_margin {0.25} def%Enter paper margin in inches
% If you would rather specify your paper sizes in millimeters, comment
% out the above three lines and uncomment the below six lines.
% Remember to change the bounding box if you change the paper parameters.
% Note ISO 216 A4 paper = 210mm x 297mm, US Letter paper = 8.5in x 11in = 215.9mm x 279.4mm
%/paper_width_in_mm {210} def                           % Enter paper width in mm
%/paper_width {paper_width_in_mm 25.4 div inch} def     % millimeters/25.4 = inches
%/paper_height_in_mm {297} def                          % Enter paper height in mm
%/paper_height {paper_height_in_mm 25.4 div inch} def   % millimeters/25.4 = inches
%/paper_margin_in_mm {6.35} def                         % Enter paper margin in mm
%/paper_margin {paper_margin_in_mm 25.4 div inch} def   % millimeters/25.4 = inches

% *** Enter your fonts here ***
/standard_font {/Helvetica-Bold} def
/italic_font {/Helvetica-Oblique} def
/italic_font_bold {/Helvetica-BoldOblique} def
```

```
%Enter name for layer caption
/layer_name (Layer) def

/dummy_string 20 string def % Empty dummy string to use when converting integers to strings

% --- Calculations ---

/number_of_layers {layer_stackup length 1 add} def %Use number of elements in layer array plus one lay
er for the thickness total

%Define height of layer stackup to be proportional to paper height minus margins
/stackup_height {paper_height 8 paper_margin mul sub} def %Stackup height in inches

% Calculate title font size as a function of the space available and fab name length
% ... but if the fab name is small, don’t let font size be greater than 36 point.
/title_font_size {paper_width paper_margin 2 mul sub 72 mul fab_name length div 1.75 mul dup 36 gt{pop
 36}if } def

% Get a smaller font for layer_name and numbers that depends upon the number of layers
/layer_font_size { number_of_layers 24 le {14}{number_of_layers 36 le {12}{10}ifelse}ifelse } def

% Calculate width of layers
/layer_width {paper_width paper_margin 2 mul sub 0.66 mul} def %Divide by a constant to give aesthetic
 layer width
/layer_height {stackup_height 8 div} def %Divide by a constant to give aesthetic layer height

/space_per_layer {stackup_height number_of_layers div} def

% Center point for writing system name
/name_center {paper_width 2 div inch paper_height inch paper_margin inch sub title_font_size sub} def

% Starting point for drawing stackup
/stackup_start {paper_margin 8 mul inch paper_margin stackup_height add inch} def

% --- Procedures ---

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% VerticalText&Show
%
% This procedure prints text vertically. It is called with the number of
% points it is to move down for each char and with the string to be shown
% on the top of stack
%
% Example: number_of_points (String) VerticalText&Show
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/VerticalText&Show
{
        exch %Exchange number of points to move down and string
        /move_down exch def %Put number of points to move down in move_down variable
        {
                %Integer value of each char of string on top of stack in turn
                1 string        %Create a one element string
                dup             %Create a reference to the one element string that will be consumed by
 put
                0               %Index for put command
                4 -1 roll       %Put integer value of char on top of stack followed by index
                put             %Put the character to be shown into a string
                0 move_down neg rmoveto %Move down size of font points
                gsave
                dup %Make a copy of string since stringwidth is destructive

                %Get X and Y width of string, get rid of Y, divide X
                %by 2, make it negative and move down that much
                stringwidth pop 2 div neg 0 rmoveto
                show
                grestore
        } forall
} def
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CenterText&Show
%
% This procedure is called with the string to be centered on the top of stack
% followed by the x and y position it is to be centered on
%
% Example: x_position y_position (String) CenterText&Show
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/CenterText&Show
{
    dup         %Make a copy of the string to be centered
    stringwidth %Get the width of the string
    pop         %Discard y component of string width
    2 div       %Find x/2 ( 1/2 string width )
    4 -1 roll exch sub  %Decrement the x position by 1/2 the string width
    3 -1 roll moveto    %Move to correct spot
    show        %Show string
} def

% --- Start program ---
newpath %Start with a clean slate

% --- Draw the system ---
standard_font findfont title_font_size scalefont setfont %Get a nice big font

% Move to the point we want the name centered on and show it
name_center fab_name CenterText&Show

% Show layer_name vertically
standard_font findfont layer_font_size scalefont setfont %Get a smaller font
0 setgray               %Make font the maximum darkness
stackup_start moveto    %Move to stackup start
% Move right enough and up enough for vertical layer_name
(1) stringwidth pop % Calculate width of an arbitrary digit
-1 mul % Multiply by -1 to move right in x direction
layer_name length layer_font_size mul rmoveto %Move up (y direction) enough for layer name to display
properly
layer_font_size layer_name VerticalText&Show %Write layer_name down vertically

stackup_start moveto %Move to stackup start

% This forall loop prints out the layers from top to bottom
layer_stackup
{
    0 space_per_layer neg inch rmoveto %Move down the space allotted for one layer

    % Write layer # at the layer edge
    gsave
    currentpoint translate %Move origin to current position
    standard_font findfont layer_font_size scalefont setfont %Get a smaller font
    0 setgray %Make font the maximum darkness
    dup %Save layer parameters for next operation
    1 get %Get layer # string on stack
    dup stringwidth pop -1 mul 0 rmoveto %Move left depending upon how big string is
    currentpoint %Save the current point
    3 -1 roll %Get string on top of stack, followed by current point
    CenterText&Show %Center the string at the current point and show
    grestore

    % Write Cu weight
    gsave
    currentpoint translate %Move origin to current position
    italic_font findfont layer_font_size 1 sub scalefont setfont
    0 setgray %Make font the maximum darkness
    dup %Save layer parameters for next operation
    2 get %Get Cu weight on stack
    % If the string isn't null
```

```
dup () ne {
          paper_margin -3.5 mul inch 0 rmoveto %Move to print location
          show
          cu_weight_units
          show
     }
     {
         pop % If the string is null, don't show anything.
     } ifelse
grestore

% Write layer thickness
gsave
currentpoint translate %Move origin to current position
italic_font findfont layer_font_size 1 sub scalefont setfont
0 setgray %Make font the maximum darkness
dup %Save layer parameters for next operation
3 get %Get layer thickness on stack
% If the string isn't null
dup () ne {
        paper_margin -7 mul inch 0 rmoveto %Move to print location
        show
        layer_thickness_units
        show
     }
     {
         pop % If the string is null, don't show anything.
     }
     ifelse
grestore

% Draw a sheared rectangle
gsave
currentpoint translate %Move origin to current position
dup %Save layer parameters for next operation
4 get %Get gray value on stack
% If gray value is null, make it 1.0
dup () eq {
           pop
          1.0
     }
     {
        cvr % Convert gray value from a string to a real number
     }
     ifelse
setgray %Make rectangle this shade of gray
[ 1 0 45 sin 1 0 0 ] concat
0 0 %Lower left corner of rectangle is origin
layer_width inch layer_height inch %Rectangle is this wide and high
rectfill %Draw a filled rectangle
grestore

% Draw a line around the sheared rectangle
gsave
currentpoint translate %Move origin to current position
[ 1 0 45 sin 1 0 0 ] concat
0 0 %Lower left corner of rectangle is origin
layer_width inch layer_height inch %Rectangle is this wide and high
0 setlinewidth %Make the line as thin as possible
0 setgray %Make line the maximum darkness
rectstroke %Draw rectangle outline
grestore

% Write name on layer after layer is drawn
gsave
currentpoint translate %Move origin to current position
standard_font findfont layer_font_size scalefont setfont
0 setgray %Make font the maximum darkness
0 get %Get name string on stack
% Move to the point we want the name centered on and show it
```

```
    layer_width 2 div layer_height 45 sin mul add inch layer_height inch layer_font_size sub moveto
    currentpoint %Save the current point
    3 -1 roll %Get string on top of stack, followed by current point
    CenterText&Show %Center the string at the current point and show
    grestore

}forall

% Add up layer thicknesses and display total thickness
0 % Start with a zero thickness on the stack
layer_stackup
{
        3 get%Get layer thickness
        % If thickness is not null
        dup () ne {
                    cvr %Convert to a real number
                    add
        }
        {
            pop % If a thickness is null, discard it
        } ifelse
} forall
% If the total thickness is greater than zero, display it
dup 0 gt {
        0 space_per_layer neg inch rmoveto %Move down the space allotted for one layer
        dummy_string cvs paper_margin -7 mul inch 0 rmoveto
        italic_font_bold findfont layer_font_size scalefont setfont
        show %Display layer thickness
        layer_thickness_units show %Display units
        ( total) show %Display "total" string
}
{
    pop %Pop unused zero total off stack
} ifelse

% showpage is permitted in an EPS file; the application importing
% an EPS file is responsible for redefining it
showpage
%%EOF
```