The Penultimate Land Pattern Naming Convention

John C. Luciani Jr.

January 17, 2007

1 IPC 7351 Issues

- Four different naming conventions for connectors.
  1. Header connectors with 100mil center pin centers have the name format `HDR`⟨*rows*⟩`x`⟨*pins per row*⟩.
  2. Connectors from AMP and Molex use the naming convention
     ⟨*series number*⟩`-`⟨*pin quantity*⟩
  3. Connectors from BERG, CUI-STACK, HIROSE, JST, KYCON, SAMTEC, SWITCHCRAFT use the manufacturer part number as the land pattern name.
  4. Connectors from manufacturers, other than the ones previously listed, use the naming convention
     ⟨*manufacturer*⟩`_`⟨*manufacturer part number*⟩

  This naming convention breaks when two (or more) of the manufacturers listed create identical part numbers.
- Common SMD package names (402, 603, 805, etc) for resistors, capacitors, and inductors are not used. To use these package sizes the land pattern names would require a prefix such as CAPC, CAPCP, RES, etc. Since Width and length specifications in IPC-7351 are in mm/100 — 402, 603, 805 would change to 102x51, 152x76, 203x127. Having to embed component function in the land-pattern name prevents re-use of these package styles.
- It is difficult to discern the component specifications from the land pattern name. Some specifications have suffixes most do not.

2 The Penultimate Naming Convention

2.1 Advantages

- Manufacturer specifications can be attached to any land-pattern using a suffix.
- Connector numbering is consistent.
- Each specification field has an identifier suffix for unambiguous specification of a field value.

2.2 Open Issues

- Specifications for Level A, B, C material conditions have not been defined.
- General suffixes are not implemented. These suffixes include `VIA`, `_HS`, `_BEC`, `_SGD` and `_213`.
- Add mounting orientation to the footprint name specification.

3 Syntax

⟨*footprint name*⟩ := ⟨*component group identifier*⟩(`-` ⟨*component specification*⟩)?(`__`⟨*manufacturer data*⟩)?

⟨*component group identifier*⟩ := ⟨*component group name*⟩ (`_`⟨*component subgroup name*⟩)⋆
⟨*component group name*⟩ := (⟨*upper case letter*⟩| ⟨*digit*⟩)+
⟨*component subgroup name*⟩ := (⟨*upper case letter*⟩| ⟨*digit*⟩)+

⟨*component specification*⟩ := ⟨*number*⟩? ⟨*specification identifier*⟩
⟨*specification identifier*⟩ := ⟨*string*⟩

⟨*manufacturer data*⟩ := ⟨*manufacturer name*⟩ (`_`⟨*manufacturer component data*⟩)?
⟨*manufacturer component data*⟩ := ⟨*package code*⟩ | ⟨*component series*⟩ | ⟨*part number*⟩

⟨*package code*⟩ := ⟨*string*⟩ (`-`⟨*string*⟩)⋆`_Package`
⟨*component series*⟩ := ⟨*string*⟩ (`-`⟨*string*⟩)⋆`_Series`
⟨*part number*⟩ := (⟨*letter*⟩ | ⟨*digit*⟩)+

⟨*string*⟩ := ⟨*letter*⟩ (⟨*letter*⟩ | ⟨*digit*⟩)+
⟨*number*⟩ := ⟨*digit*⟩+

⟨*letter*⟩ := ⟨*upper case letter*⟩ | ⟨*lower case letter*⟩
⟨*upper case letter*⟩ := `A` | `B` | `C` | ... | `Z`
⟨*lower case letter*⟩ := `a` | `b` | `c` | ... | `z`
⟨*digit*⟩ := `0` | `1` | `2` | `3` | `4` | `5` | `6` | `7` | `8` | `9`

3.1 Conventions

- Distance measurements are in hundreths of a mm.
- With pin one in the upper left corner length is in the X direction and width is in the Y direction.
- Lead span 1 (`L1`) is along the length and lead span 2 (`L2`) is along the width.
- Chip Capacitors, Resistors and Inductors (RES, CAP and IND) Pin 1 (Positive) on Left
- Molded Inductors (INDM), Resistors (RESM), Tantalum Capacitors (CAPT) Pin 1 (Positive) on Left
- Precision Wire-wound Inductors Pin 1 (Positive) on Left
- MELF Diode Pin 1 (Cathode) on Left
- SOD Diodes Pin 1 (Cathode) on Left
- Aluminum Electrolytic Capacitors Pin 1 (Positive) on Left
- SOT Devices (SOT23, SOT23-5, SOT223, SOT89, SOT143, etc.) Pin 1 Upper Left
- TO252 & TO263 (DPAK Type) Devices Pin 1 Upper Left
- Small Outline Gullwing ICs (SOIC, SOP, TSOP, SSOP, TSSOP) Pin 1 Upper Left
- Ceramic Flat Packs (CFP) Pin 1 Upper Left
- Small Outline J Lead ICs (SOJ) Pin 1 Upper Left
- Quad Flat Pack ICs (PQFP, SQFP) Pin 1 Upper Left
- Ceramic Quad Flat Packs (CQFP) Pin 1 Upper Left

- Bumper and Plastic Quad Flat Pack ICs (BQFPC, PQFPC Pin 1 Center) Pin 1 Top Center
- Plastic Leaded Chip Carriers (PLCC) Pin 1 Top Center
- Leadless Chip Carriers (LCC) Pin 1 Top Center
- Leadless Chip Carriers (LCCS Pin 1 on Side) Pin 1 Upper Left
- Quad Flat No-Lead ICs (QFN) QFNS & QFNRV, QFNRH Pin 1 Upper Left
- Ball Grid Arrays (BGA) Pin A1 Upper Left

3.2 Specification Indentifier

Each specification field contains one of the specification identifiers listed in Table 1.

Identifier	Description	Type
P	pin_spacing	dimension
C	pin_columns	count
R	pin_rows	count
N	pin_count	count
d	pin_diameter	dimension
H	height	dimension
L	body_length	dimension
W	body_width	dimension
D	body_diameter	dimension
L1	lead_span_1	dimension
L2	lead_span_2	dimension
WT	thermal_pad_width	dimension
LT	thermal_pad_length	dimension
TP	thermal_pad	boolean
PS	pad_size	dimension
MS	mask_size	dimension

Table 1: Specification Identifiers

3.3 Package Specification Fields

Each specification field that is required to create a land pattern name for a particular package is listed in Table 2. I am in the process of adding all of the package types listed in Table 3 and Table 4.

Name	Fields
TSSOP	pin_spacing, lead_span_1, pin_count
HTSSOP	pin_spacing, lead_span_1, pin_count, thermal_pad_length, thermal_pad_width
QFN	pin_spacing, body_width, body_length, pin_count, thermal_pad, thermal_pad_width, thermal_pad_length
QFP	lead_span_1, lead_span_2, pin_count
CAPR	pin_spacing, body_diameter, body_length, pin_diameter
CAPPR	pin_spacing, body_diameter, body_length, pin_diameter
SOIC	pin_spacing, lead_span_1, pin_count
CAPA	pin_spacing, body_diameter, pin_diameter

Table 2: Package Specification Fields

3.4 Component Groups

Component Groups

Termination	Description	Component Group Identifier
SMD	Ball Grid Array	BGA
SMD	Ball Grid Array'sw/Staggered Pins	SBGA
SMD	Capacitiors, Chip, Non-polarized	CAPC
SMD	Capacitiors, Chip, polarized	CAPCP
SMD	Capacitiors, Chip, Wire Rectangle	CAPCWR
SMD	Capacitiors, Molded, Non-polarize	CAPM
SMD	Capacitiors, Molded, polarize	CAPMP
SMD	Capacitiors, Aluminum Electrolutic	CAPAE
SMD	Ceramic Flat Packages	CFP
SMD	Column Grid Arrays	CGA
SMD	Diodes, Molded	DIOM
SMD	Diodes, MELF	DIOMMELF
SMD	Inductors, Chip	INDC
SMD	Inductors, Molded	INDM
SMD	Inductors, Percision Wire Wound	INDP
SMD	Plastic Leaded Chip Carriers Square	PLCC
SMD	Plastic Leaded Chip Carriers Rectangular	PLCCR
SMD	Plastic Leaded Chip Carrier Sockets Square	PLCCS
SMD	Plastic Leaded Chip Carrier Sockets Rectangular	PLCCRS
SMD	Plastic Quad Flat Packages	PQFPS
SMD	Plastic Quad Flat Packages	PQFPC
SMD	Bumper Quad Flat Packages	BQFPS
SMD	Bumper Quad Flat Packages	BQFPC
SMD	Quad Flat Packages	QFP
SMD	Shrink Quad Flat Packages	SQFP
SMD	Thin Quad Flat Packages	TQFP
SMD	Thin Quad Flat Packages	TSQFP
SMD	Ceramic Quad Flat Packages	CQFP
SMD	Quad Flat No Lead Packages	QFN
SMD	Quad Leadless Ceramic Chip Carriers	LCC
SMD	Quad Leadless Ceramic Chip Carriers	LCCS
SMD	Quad Bottom Chip Carrier	QBCC
SMD	Resistors, Chip	RESC
SMD	Resistors, Molded	RESM
SMD	Resistors, MELF	RESMELF
SMD	Small Outline IC	SOJ
SMD	Small Outline Intergrated Circuit	SOIC
SMD	Small Outline Packages	SOP
SMD	Shrink Small Outline Packages	SSOP
SMD	Thin Small Outline Packages	TSOP
SMD	Thin Shrink Small Outlind Packages	TSSOP
SMD	Very Small Outline Packages	VSOP
SMD	SOD	SON
SMD	SOT89	SOT89
SMD	TO	TO

Component Groups (continued)

Termination	Description	Component Group Identifier
SMD	Amplifiers	AMP
SMD	Batteriers	BAT
SMD	Capacitors, Variable	CAPV
SMD	Capacitors, Chip, Array, Concave	CAPCAV
SMD	Capacitors, Chip, Array, Convex	CAPCAX
SMD	Capacitors, Chip, Array, Flat	CAPCAF
SMD	Capacitors, Miscellaneous	CAP
SMD	Crystals	XTAL
SMD	Diodes, Miscellaneous	DIO
SMD	Diodes, Bridge Rectifiers	DIOB
SMD	Ferrite Beads	FB
SMD	Fiducials	FID
SMD	Filters	FIL
SMD	Fuses	FUSE
SMD	Fuse, Resettable	FUSER
SMD	Inductors, Miscellaneous	IND
SMD	Inductors, Chip, Array, Concave	INDCAV
SMD	Inductors, Chip, Array, Convex	INDCAX
SMD	Inductors, Chip, Array, Flat	INDCAF
SMD	Keypad	KEYPAD
SMD	LEDS	LED
SMD	Liquid Crystal Display	LCD
SMD	Microphones	MIC
SMD	Opto Isolators	OPTO
SMD	Oscillators	OSC
SMD	Potentiometers	POT
SMD	Resistors, Chip, Array, Concave	RESCAV
SMD	Resistors, Chip, Array, Convex	RESCAX
SMD	Resistors, Chip, Array, Flat	RESCAF
SMD	Relays	RELAY
SMD	Speakers	SPKR
SMD	Switches	SW
SMD	Test Points, Round	TP
SMD	Test Points, Square	TPS
SMD	Thermistors	THERM
SMD	Transducers	XDCR
SMD	Transient Voltage Suppressors	TVS
SMD	Transient Voltage Suppressors, Polarized	TVSP
SMD	Transistor Outlines, Custom	TRANS
SMD	Transformers	XFMR
SMD	Trimmers	TRIM
SMD	Tuners	TUNER
SMD	Varistors	VAR
SMD	Voltage Controlled Oscillators	VCO
SMD	Voltage Regulators, Custom	VREG

Component Groups

Termination	Description	Component Group Identifier
TH	Capacitors, Non Polarized Axial	CAPA
TH	Capacitors, Non Polarized Radial, Round	CAPR
TH	Capacitor, Polarized Radial	CAPPR
TH	Converters	CONV
TH	Crystals	XTAL
TH	Diodes	DO
TH	Dual-In-Line Packages	DIP
TH	Dual-In-Line Sockets	DIPS
TH	Ferrite Beads	FB
TH	Headers, .100 inch Pin Centers	HDR
TH	Heat Sinks	HSINK
TH	Jumpers, Wire	JMP
TH	Mounting Holes Nonplated	MTG
TH	MOV	MOV
TH	PAD	PAD
TH	Photo Detectors	PHODET
TH	Pin Grid Arrays	PGA
TH	Potentiometers	POT
TH	Regulators	REG
TH	Resistors, Axial Leads	RES
TH	Resistor Networks	SIP
TH	Shield, Off the shelf	SHEILD
TH	Stiffners	STIF
TH	Transistor Outlines	TO
TH	Trimmers	TRIM

4 Examples

`QFN-50P-700W-700L-48N-500WT-500LT__LTC`	Quad Flatpak No leads 0.5mm pitch 7x7mm body 48 pins 5x5mm thermal pad
`CON_FPC-50P-30R-1C`	Flexible Printed Circuit Connector, 0.5mm pitch, 30 rows, 1 column
`CAPC-570L-500W__Murata_GRM55_Series`	Ceramic capacitor 5.7x5mm, Murata GRM55 Series
`0402__Murata`	0404 package from Murata
`2220__Murata_GRM55_Series`	2220 package from Murata GRM55 series
`CON_USB_TYPEB__Keystone_924`	Type B USB Connector Keystone part number 924
`HTSSOP-65P-640L1-16N__LTC_FE_Package`	HTSSOP package (TSSOP with a thermal pad), 0.65mm pitch, 6.4mm lead space 16 pins defined by the Linear Technology FE specification.
`TSSOP-65P-640L1-8N`	TSSOP package, 0.65mm, 6.4mm lead span, 8 pins
`CAPA-150P-400D-45d`	Axial lead capacitor, 1.5mm pitch, 4mm body diameter, 0.45mm lead diameter

Table 5: Footprint Name Examples

5 Perl Module

The Perl module in Listing 1 contains code to build footprint name strings from component specifications in a hash and code to build a hash containing the footprint specifications embedded in a footprint name string.

Listing 1: Footprint Package

```
# Copyright (C) 2005 John C. Luciani Jr.

# This program may be distributed or modified under the terms of
# version 0.1 of the No-Fee Software License published by
# John C. Luciani Jr.

# The Perl module contains two exported subroutines --- footprint_name
# and parse_footprint_name. footprint_name creates a footprint name
# string from values in a hash. parse_footprint_name populates a hash
# with values parsed from a footprint name string.

package         Footprint_name_0;

require         Exporter;
use vars        qw($VERSION @ISA @EXPORT @EXPORT_OK %EXPORT_TAGS);
@ISA          = qw(Exporter);
@EXPORT       = qw(footprint_name parse_footprint_name);
@EXPORT_OK    = qw();

%EXPORT_TAGS = qw();

use strict;
use warnings;
use Carp;
use Tie::IxHash;
use Data::Dumper;

my %Field;
my %Pkg_fields;
my %Field_type;
my %Opt_field;  # optional field names

# _parse_field_rec populates thet %Field and %Field_type hashes from
# the lines of a here-doc.

sub _parse_field_rec  {
    my $str = shift;
    return unless $str =~ s/(\S+)\s+(\S+)\s*//;
    my ($abbrev, $name) = ($1, $2);
    $Field{$abbrev} = $name;
    $Field{$name} = $abbrev;
    $Field_type{$name} = $1 if $str =~ /(dimension boolean count)/;
}

# Populate %Field %Field_type and %Pkg_fields

BEGIN
{
    tie %Field, "Tie::IxHash";

    # Each field that is used in a land pattern name is listed
    # below. To add a new field place a line containing the
    # field suffix, hash-key, and field type in the following
    # here-doc.

    map { &_parse_field_rec($_) } <<'    END_FIELDS' =~ /(.+)/g;
    P   pin_spacing          dimension
    Sr  pin_row_spacing      dimension
    C   pin_columns          count
    R   pin_rows             count
    N   pin_count            count
    d   pin_diameter         dimension
    H   height               dimension
```

```
64      L   body_length          dimension
65      W   body_width           dimension
66      D   body_diameter        dimension
67      L1  lead_span_1          dimension
68      L2  lead_span_2          dimension
69      WT  thermal_pad_width    dimension
70      LT  thermal_pad_length   dimension
71      TP  thermal_pad          boolean
72      END_FIELDS
73
74
75      # Each footprint name is built using the fields defined below.
76
77      %Pkg_fields = map { ($1 => [split /\s+/, $2]) if /(\S+)\s+(.*)/ } <<'
            END_PKG_FIELDS' =~ /(.+)/g;
78      SSOP    pin_spacing lead_span_1   pin_count
79      TSSOP   pin_spacing lead_span_1   pin_count
80      OPTO    pin_spacing lead_span_1   pin_count
81      MSOP    pin_spacing lead_span_1   pin_count
82      TSOP    pin_spacing lead_span_1   pin_count
83      HTSSOP pin_spacing lead_span_1   pin_count   thermal_pad_length thermal_pad_width
84      QFN     pin_spacing body_width    body_length   pin_count   thermal_pad
            thermal_pad_width thermal_pad_length
85      LQFP    pin_spacing lead_span_1   lead_span_2   pin_count
86      QFP     pin_spacing lead_span_1   lead_span_2   pin_count
87      SQFP    pin_spacing lead_span_1   lead_span_2   pin_count
88      CAPR    pin_spacing body_diameter body_length pin_diameter
89      CAPPR   pin_spacing body_diameter body_length pin_diameter
90      SOIC    pin_spacing lead_span_1   pin_count
91      CAPA    pin_spacing body_diameter pin_diameter
92      SOD123 lead_span_1 body_width
93      SOD323 lead_span_1 body_width
94      SMA     lead_span_1 body_width
95      SMB     lead_span_1 body_width
96      SMC     lead_span_1 body_width
97      END_PKG_FIELDS
98
99      # Fields in %Opt_field are considered to be optional.
100
101     %Opt_field = (map { $_ => 1 } qw(thermal_pad thermal_pad_length thermal_pad_width
            ));
102
103 }
104
105
106 sub parse_footprint_name ($\%) {
107     my $name = shift; # footprint name to parse
108     my $ref  = shift; # reference to the hash that will receive the field values
109
110     # The string following the two underscores contains a manufacturer
111     # name and optionally a part description. If a manufacturer part
112     # description is present it is preceeded by a single underscore.
113
114     if ($name =~ s/__(.*)$//) {
115         $ref->{manufacturer} = $1;
116         $ref->{manufacturer_description} = $1
117             if $ref->{manufacturer} =~ s/_(.*)//;
118     }
119     if ($name =~ s/^([^-]+)//) {
120         $ref->{component_group} = $1;
121         $ref->{component_subgroup} = $1
122             if $ref->{component_group} =~ s/_(.*)$//;
123     }
124     while ($name =~ m/-?(\d*)([A-Za-z]?[^-]*)/g) {
125         next if $1 eq '' && $2 eq '';
126         if ($2 eq '') {
127             $ref->{ package_name } = $1;
128         } elsif (defined $Field{$2}) {
129             $ref->{ $Field{$2} } = $1;
130             if ($Field_type{ $Field{$2} } eq 'dimension') {
131                 $ref->{ $Field{$2} } /= 100; # hundreths of a mm
132             } elsif ($Field_type{ $Field{$2} } eq 'boolean') {
133                 $ref->{ $Field{$2} } = 1;
```

```
134             }
135         } else {
136             carp "Field $2 is not defined";
137         }
138     }
139 }
140
141 sub footprint_name (\%) {
142     my ($ref) = @_;
143     my $name;
144     if (defined $ref->{package_name}) {
145         $name = $ref->{package_name};
146     } else {
147         $name = &footprint_str(map { $ref->{$_} } qw(component_group
            component_subgroup));
148         my @defined_fields = defined $Pkg_fields{$name}
149                                 ? @ { $Pkg_fields{$name} }
150                                 : grep { defined $ref->{$_} } keys %Field;
151         foreach my $field (@defined_fields) {
152             if (!defined $ref->{$field} && $Opt_field{$field}) {
153                 # an optional field is missing
154             } elsif ($Field_type{$field} eq 'dimension') {
155                 my $value = $ref->{$field};
156                 carp "Value was not defined for field $field" unless defined $value;
157                 $value *= 25.4/1000
158                     if defined $ref->{units_of_distance} && $ref->{units_of_distance}
                        =~ /mil/;
159                 $name .= sprintf("-%.0f%s", $value * 100, $Field{$field});
160             } elsif ($Field_type{$field} eq 'boolean') {
161                 # no value for a boolean field
162                 $name .= sprintf("-%s", $Field{$field}) if $ref->{$field};
163             } else {
164                 $name .= sprintf("-%s%s", $ref->{$field}, $Field{$field});
165             }
166         }
167     }
168     my $mfgstr =  &mfg_str($ref);
169     $name .= "__$mfgstr" unless $mfgstr eq '';
170     return($name);
171 }
172
173
174 sub mfg_str ($) {
175     my ($ref) = @_;
176     my $desc = $ref->{manufacturer_part_number}    $ref->{manufacturer_description};
177     if (defined $desc) {
178     } else {
179         my %suffix = qw(manufacturer_package_code Package component_series Series);
180         foreach (keys %suffix) {
181             next unless defined $ref->{$_};
182             next if $ref->{$_} eq '';
183             $desc = sprintf("%s_%s", $ref->{$_}, $suffix{$_});
184             last;
185         }
186     }
187     &footprint_str( $ref->{manufacturer}, $desc);
188 }
189
190
191
192 sub footprint_str ($$) {
193     my ($s1, $s2) = @_;
194     return ''        unless defined $s1;
195     return $s1 unless defined $s2;
196     return $s1 . '_' . $s2;
197 }
198
199
200 1;
201
202
203 # Style (adapted from the Perl Cookbook, First Edition, Recipe 12.4)
204
```

```
# 1. Names of functions and local variables are all lowercase.
# 2. The module's persistent variables (either file lexicals
#    or package globals) are capitalized.
# 3. Identifiers with multiple words have each of these
#    separated by an underscore for readability.
# 4. Constants are all uppercase.
# 5. If the arrow operator (->) is followed by either a
#    method name or a variable containing a method name then
#    there is a space before and after the operator.
# 6. Function names, variable names, hash keys that are meant
#    to be used only within the current package have an
#    underscore prefix.


__END__
[test program]
#!/usr/bin/perl

use strict;
use warnings;
use Carp;
use Data::Dumper;

use Footprint_name_0;

while (<DATA>) {
    s/\#.*//; # Remove comments
    s/^\s*//; # Remove leading spaces
    s/\s*$//; # Revove trailing spaces
    next unless length; # Skip empty lines
    my %c;
    &parse_footprint_name($_, \%c);
    print("--------------------\n");
    printf("%s\n%s\n", $_, &footprint_name(\%c));
    print "\%c = ", Dumper(\%c); ### DeBuG ###
}
print("--------------------\n");

my %C = (units_of_distance => 'mm',
         component_group => 'TSSOP',
         thermal_pad => 1,
         pin_spacing => 0.65,
         lead_span_1 => 6.4,
         pin_count => 16 );

foreach my $key (keys %C) {
    print "$key = $C{$key}\n";
}
printf("%s\n", &footprint_name(\%C));


__DATA__
QFN-50P-700W-700L-48N-500WT-500LT__LTC
CON_FPC-50P-1C-30R
CAPC-570L-500W__Murata_GRM55_Series
0402__Murata
2220__Murata_GRM55_Series
CON_USB_TYPEB__Keystone_924
TSSOP-65P-640L1-16N__LTC_FE_Package
TSSOP-65P-640L1-8N
CAPA-150P-400D-45d
HTSSOP-65P-640L1-16N-294LT-358WT__LTC_FE_Package
HTSSOP-65P-810L1-28N-420LT-770WT__TI_PowerPAD_Package
```

6 Possible Changes *a.k.a. notes to myself*

6.1 Hardware Footprints

Add a component group identifier for hardware with the syntax —

⟨*component group identifier*⟩ := `HW-`⟨*subgroup name*⟩⟨*style character*⟩?
⟨*style character*⟩ := ⟨*lower case letter*⟩

Identifier	Description	Type
`SS`	screw size	dimension
`MSS`	metric screw size	dimension

Table 6: Tags for Hardware Components

Component Sub-group	Style	Description
`HEX`, `HEXWAS`	`a`	center hole is replaced with silkscreen cross hairs + pilot hole
	`b`	center hole is replaced with copper cross hairs + pilot hole

Table 7: Style Characters for Hardware Components

6.1.1 Examples

hexnut `HW-HEXa_2SS-156D` number 2 hexnut with a hex-diameter of 0.156 inches style `a`.

hexnut and washer `HW-HEXWASa_2SS_156D` number 2 hexnut with a 0.156 OD washer.

References

IPC. (2005, January). IPC-7351 Naming Convention for Standard SMT Land Patterns. (Retrieved July 7, 2005, from http://landpatterns.ipc.org/IPC-7351Table3-15January2005.pdf)