What's New in LEF/DEF 6.0

Product Version 6.0
September 2022

Contents

About This Manual

This document provides information on new and changed features for version 6.0 of the Cadence® Library Exchange Format (LEF) and Design Exchange Format (DEF) integrated circuit description languages.

How This Document Is Organized

This *What's New* document is organized into the following chapters:

■   LEF Syntax Changes

     This chapter describes features that were added or changed in the LEF syntax since version 5.8 of LEF.

     Any LEF syntax that has been made obsolete since version 5.8 are also highlighted in this chapter.

■   DEF Syntax Changes

     This chapter describes features that were added or changed in the DEF syntax since version 5.8 of DEF.

     Any DEF syntax that has been made obsolete since version 5.8 are also highlighted in this chapter.

■   ALIAS Statement

     This chapter explains why the ALIAS statement has been removed and what to do if your old files still use ALIAS statements.

Related Documents

The following document provides detailed information about LEF and DEF.

■   *LEF/DEF Language Reference*

1

LEF Syntax Changes

Overview

The major changes in the LEF/DEF 6.0 version are summarized below:

- A `PROPERTY` statement has been added to every LEF/DEF object to make LEF/DEF more extensible. When applications want to add new attributes to an object, it can be passed through LEF/DEF using a `PROPERTY` statement.

- An explicit `WIDTH` value has been added to all `NETS` routing shapes whenever it does not match the net `DEFAULT` rule or `NONDEFAULTRULE` width. This change is important for new process nodes, given wrong-width rules and the need to create artificial NDRs for odd-widths to meet electro-migration (EM) constraints, timing constraints, or double-patterning restrictions.

- Obsolete statements and keywords that have not been in use for a long time have been removed.

- Some enhancements have been made based on user requests, such as adding `NDR` and `WIDTH` to `TRACKS`, adding explicit `REAL` | `ABSTRACT` | `NOROUTE` keywords in `MACRO OBS`, and so on.

New LEF Syntax

The following sections describe the enhancements that have been made to the LEF syntax. For the complete syntax information and examples, see the "LEF Syntax" chapter in the *LEF/ DEF Language Reference*.

Version 6.0 Added

You can include a `VERSION` statement as follows in your syntax:

```
VERSION 6.0 ;
```

The `VERSION` statement is optional. Most applications default to the 5.8 version of LEF. However, a default value of 5.8 is not formally part of the language definition; therefore, you cannot be sure that all applications use this default value. The recommended practice is to specify the `VERSION` statement in the LEF file.

Property Name Prefixes Changed to LEF_CDN_xxx

The documented prefix of every `LEF58_xxx` property has been changed to `LEF_CDN_xxx`. The new Cadence-specific prefix is not tied to any particular version of LEF.

Note that Cadence applications that use these properties will still support `LEF58_xxx` property names (and `LEF57_xxx` for a few old names), but `LEF_CDN_xxx` names will be used in the documentation and examples.

New objectTypes in PROPERTYDEFINITIONS

To support `PROPERTY` statements on more objects, the list of supported *objectType* names in `PROPERTYDEFINITIONS` has been expanded. The following *objectType* names have been added:

■ `PORTOBS` - For a LEF PIN shape or via in either the `PORT` or `OBS` section.

■ `SITE` - For a LEF SITE definition.

MANUFACTURINGGRID Added to All Layer Types

The `LEF58_MANUFACTURINGGRID` property has been replaced with native syntax, although the property version is still supported for backward compatibility.

All layer types, CUT, ROUTING, IMPLANT, OVERLAP, or MASTERSLICE, may have this new statement.

```
LAYER layerName
      TYPE ROUTING ; #Or TYPE CUT, IMPLANT, MASTERSLICE, OVERLAP
      [MANUFACTURINGGRID value ;]
```

New OBSSPACING Statement in MACRO

The LEF58_OBSSPACING property has been replaced with native syntax in MACRO.

The new MACRO syntax is:

```
MACRO macroName
      [CLASS ...]
      ...
      [OBSSPACING { FULLDRC | MIN | spacing } [LAYER layer] ... ; ...
      ...
```

New PROPERTY Statement in Layer Geometries in MACRO

The MACRO adds a new PROPERTY statement in the Layer Geometries syntax as shown in bold below. This new statement can be used for OBS and PIN PORT shapes and VIAS.

```
{ LAYER layerName
      [PROPERTY {propName propVal}] ...
      [EXCEPTPGNET]
      [SPACING minSpacing | DESIGNRULEWIDTH value] ;
  [WIDTH width ;]
  { PATH [MASK maskNum] pt ... ;
  | PATH [MASK maskNum] ITERATE pt ... stepPattern ;
  | RECT [MASK maskNum] pt pt ;
  | RECT [MASK maskNum] ITERATE pt pt stepPattern ;
  | POLYGON [MASK maskNum] pt pt pt pt ... ;
  | POLYGON [MASK maskNum] ITERATE pt pt pt pt ... stepPattern ;
  } ...
| VIA
      [PROPERTY {propName propVal}] ...
      [MASK viaMaskNum] pt viaName ;
| VIA
      [PROPERTY {propName propVal}] ...
      [MASK viaMaskNum] ITERATE pt viaName stepPattern ;
} ...
```

New Keywords in Layer Geometries in MACRO for OBS Shapes

MACRO adds new `REAL`, `ABSTRACT`, and `NOROUTE` keywords in the Layer Geometries syntax as shown in bold below. These new keywords apply only to OBS shapes, and are intended to define the OBS meaning in different contexts explicitly.

The Layer Geometries syntax is used for both `OBS` and `PIN PORT` statements, but the `EXCEPTPGNET`, `SPACING`, `DESIGNRULEWIDTH`, `REAL`, `ABSTRACT` and `NOROUTE` keywords below are ignored on PIN PORT shapes because PIN PORT shapes are always considered `REAL` shapes with full DRC checks.

The syntax below allows `SPACING`, `DESIGNRULEWIDTH`, `EXCEPTPGNET`, `REAL`, `ABSTRACT`, and `NOROUTE` keywords both on shapes (`PATH`, `RECT`, `POLYGON`) and vias (`VIA`), but they apply only to shapes. There is no usage model for any of these controls on vias, so they can be silently ignored for any via.

```
[OBS
    { LAYER layerName
        [PROPERTY {propName propVal}] ...#also new syntax
        [EXCEPTPGNET]
        [REAL | ABSTRACT | NOROUTE]
        [SPACING minSpacing | DESIGNRULEWIDTH value] ;
      [WIDTH width ;]
      { PATH [MASK maskNum] pt ... ;
      | PATH [MASK maskNum] ITERATE pt ... stepPattern ;
      | RECT [MASK maskNum] pt pt ;
      | RECT [MASK maskNum] ITERATE pt pt stepPattern ;
      | POLYGON [MASK maskNum] pt pt pt pt ... ;
      | POLYGON [MASK maskNum] ITERATE pt pt pt pt ... stepPattern ;
      } ...
    | VIA [MASK maskNum] pt viaName ;
    | VIA [MASK maskNum] ITERATE pt viaName stepPattern ;
    } ...
END]
```

New PROPERTY Statement in SITE

To allow possible extensions in the future, a property is added to a SITE definition. The updated `SITE` syntax is with the change in bold is:

```
SITE siteName
    CLASS {PAD | CORE} ;
    [SYMMETRY {X | Y | R90} ... ;]
```

```
      [ROWPATTERN {previousSiteName siteOrient} ... ;]
      [PROPERTY propName propVal ;] ...
      SIZE width BY height ;
END siteName
```

Obsolete LEF Syntax

NETEXPR Removed from LEF MACRO PIN

The `NETEXPR` statement in the MACRO PIN syntax is obsolete and has been removed because it is no longer in use. The current practice is to use IEEE1801 for digital designs.

The removed syntax in shown in red is below:

```
[PIN pinName
    [TAPERRULE ruleName ;]
    [DIRECTION {INPUT | OUTPUT [TRISTATE] | INOUT | FEEDTHRU} ;]
    [USE { SIGNAL | ANALOG | POWER | GROUND | CLOCK } ;]
    [NETEXPR "netExprPropName defaultNetName" ;]
    [SUPPLYSENSITIVITY powerPinName ;]
    [GROUNDSENSITIVITY groundPinName ;]
    ...
```

LEF58_OBSSPACING Property Removed

The `LEF58_OBSSPACING` property shown below is obsolete:

```
PROPERTY LEF58_OBSSPACING
"OBSSPACING { FULLDRC | MIN | spacing } [LAYER layer]...
;... " ;
```

This property has been replaced with native syntax in MACRO. See the New OBSSPACING Statement in MACRO section for details.

LEF58_MANUFACTURINGGRID Property Removed

The `LEF58_MANUFACTURINGGRID` property shown below is obsolete:

```
PROPERTY LEF58_MANUFACTURINGGRID
"MANUFACTURINGGRID value
   ; " ;
```

This property has been replaced with native syntax in LEF Layers. See the MANUFACTURINGGRID Added to All Layer Types section for details.

2

DEF Syntax Changes

New DEF Syntax

The following sections describe the enhancements that have been made to the DEF syntax. For the complete syntax information and examples, see the "DEF Syntax" chapter in the *LEF/ DEF Language Reference*.

Version 6.0 Added

You can include a VERSION statement as follows in your syntax:

```
VERSION 6.0 ;
```

The VERSION statement is optional. Most applications default to the 5.8 version of DEF. However, a default value of 5.8 is not formally part of the language definition; therefore, you cannot be sure that all applications use this default value. The recommended practice is to specify the VERSION statement in the DEF file.

New objectTypes in the PROPERTYDEFINITIONS Section

To support PROPERTY statements on more objects, the list of supported *objectType* names in PROPERTYDEFINITIONS has been expanded. The following *objectType* names have been added:

- BLOCKAGE - For a blockage shape.

- PIN - For DEF PIN properties, such as the logical PIN object.

- PINSHAPE - For individual DEF PIN shapes or vias.

- ROUTE - For routing shapes and vias in NETS.

- SCANCHAIN - For scan chains and ordered sub-chains.

- SPECIALROUTE - For SPECIALNETS and routing shapes and vias in FILLS.

- VIA - For a via definition, such as a DEF VIAS section. A VIA instance would use ROUTE or SPECIALROUTE.

- TRACKS - For tracks.

In addition, the existing COMPONENTPIN object type now applies only to PINPROPERTY in COMPONENT, and not DEF PINs, in DEF 6.0.

New Attributes and PROPERTY for Routing Blockages

New routing blockage attributes and usage have been defined in the DEF 6.0 version. In addition, PROPERTY has now been added to allow future updates to the blockage object. The updated syntax for routing blockages is shown below with the additions in bold.

Note: DESIGNRULEWIDTH and SLOTS have been made obsolete in routing blockages. See Obsolete DEF Syntax for details.

```
[BLOCKAGES numBlockages ;
[- LAYER layerName
      [+ COMPONENT compName | + FILLS | + PUSHDOWN
          | + EXCEPTPGNET | + ONLYPGNET]
      [+ SPACING minSpacing]
      [+ MASK maskNum]
      [+ NAME name]
      [+ PARTIAL density]
      [+ PROPERTY propName propVal] ...
          {RECT pt pt | POLYGON pt pt pt ...} ...
;] ...
```

New Attributes and PROPERTY for Placement Blockages

New placement blockage attributes and usage have been defined in the DEF 6.0 version. The syntax for existing SOFT and PARTIAL keywords has been modified, and new NAME, ONLYBLOCKS, and NOFLOPS keywords are added. In addition, PROPERTY has now been added to allow future updates to the blockage object. PROPERTY must be defined in the PROPERTYDEFINITIONS section for a BLOCKAGE object.

The updated syntax for placement blockages is shown below with the changes in bold.

```
[BLOCKAGES numBlockages ;
   [- PLACEMENT
        [+ NAME name]
        [+ PROPERTY propName propVal] ...
        [+ ONLYBLOCKS | + SOFT maxDensity
            | + PARTIAL maxDensity [NOFLOPS] ]
        [ + PUSHDOWN]
        [ + COMPONENT compName]
        {RECT pt pt} ...
   ;] ...
```

New Attributes in the GROUPS Section

The new semantics for GROUPS are:

■ A group is a list of hinsts, insts, and groups.

■ A specific inst can only be in one group's inst list.

■ A specific hinst can only be in one group's hinst list.

■ A group can only be in one group's group list, and "loops" are not allowed. For example, if gp1 is in gp2, and gp2 is in gp3, then gp3 cannot be in gp1.

■ An inst inside an hinst can be in a different group than the hinst. The placer and DB can handle this.

Previously, the GROUPS syntax allowed a *compNamePattern* as an approximation for hierarchical netlist names, such as hinst1/hinst2/*, and individual inst names. In DEF 6.0, this usage is replaced with explicit GROUPS, HINSTS, and COMPONENTS statements. If a group is defined as part of a power domain created by IEEE 1801 rules, it additionally has the POWERDOMAIN keyword. The updated syntax for GROUPS is shown below with the changes in bold:

```
[GROUPS numGroups ;
[- groupName
    [+ POWERDOMAIN]
    [+ GROUPS group1 [group2] ...]
    [+ HINSTS hinst1 [hinst2] ...]
    [+ COMPONENTS comp1 [comp2] ...]
    [+ REGION regionName]
    [+ PROPERTY {propName propVal} ...] ...
;] ...
END GROUPS]
```

SOFTFIXED, PHYSICAL, and PINPROPERTY Added to COMPONENTS

The following changes have been made to the syntax of the COMPONENTS section:

■ [+ SOURCE {NETLIST | DIST | USER | TIMING}] statement is obsolete. The only old usage still needed is + SOURCE DIST, which has now been replaced by + PHYSICAL. See Obsolete DEF Syntax for details.

■ A new placement status, SOFTFIXED, has been added.

■ The separate PINPROPERTIES section in the DEF file has been replaced by adding PINPROPERTY directly to the COMPONENTS section.

■ The `WEIGHT` and `REGION` keywords are obsolete. See Obsolete DEF Syntax for details.

Here's the updated syntax for the `COMPONENTS` section with the additions highlighted in bold:

```
COMPONENTS numComps ;
[- compName modelName
    [+ EEQMASTER macroName]
    [+ UNPLACED | {FIXED | SOFTFIXED | COVER | PLACED} pt orient]
    [+ MASKSHIFT shiftLayerMasks]
    [+ HALO [SOFT] left bottom right top]
    [+ ROUTEHALO haloDist minLayer maxLayer]
    [+ PINPROPERTY pinName propName propVal] ...
    [+ PHYSICAL]
    [+ PROPERTY {propName propVal} ...]...
;] ...
END COMPONENTS
```

New PROPERTY Statement in the PINS Section

The `PINPROPERTIES` section has been removed as it was obsolete, and replaced by the `DEF PIN PROPERTY` or `DEF COMPONENT PINPROPERTY` statements.

The addition to the `PINS` syntax is shown in bold below:

```
[PINS numPins ;
    [ - pinName + NET netName
      [+ SPECIAL]
      [+ DIRECTION {INPUT | OUTPUT | INOUT | FEEDTHRU}]
      [+ PROPERTY propName propVal] ...
      [+ SUPPLYSENSITIVITY powerPinName]
      [+ GROUNDSENSITIVITY groundPinName]
      ...
```

Other Enhancements in the PINS Section

The following additional enhancements have been made to the `PINS` section:

■ Previously, in DEF 5.8, the DEF PIN syntax defined a PORT with layer shapes and vias, and then translated the entire group of shapes using an origin and orient value. In DEF 5.8, the PIN PORT coordinates of `LAYER`, `POLYGON`, and `VIA` were considered as relative to the coordinates of the `FIXED`, `COVER`, and `PLACED` statements (including coordinate rotation). However, this was different from other DEF geometric data and could confuse DEF users. In DEF 6.0, the PIN syntax has been changed to use absolute

coordinates like other DEF routing and blockage shapes use, so the `FIXED`, `COVER`, and `PLACED` statements have no coordinates or *orient* values anymore. The `LAYER`, `POLYGON`, and `VIA` statements now use absolute coordinates. In addition, the VIA syntax was enhanced with an *orient* value.

■   Specific PIN shapes or vias can now have a `PROPERTY` statement.

■   The `SPACING` and `DESIGNRULEWIDTH` statements are obsolete and have been removed (see the Obsolete DEF Syntax section for details).

■   The `COVER`, `FIXED`, and `PLACED` statements no longer have *pt orient* values, like DEF 5.8 had as shown here:

```
[+ COVER pt orient | FIXED pt orient | PLACED pt orient]
```

The updated `PINS` syntax is as follows:

```
[PINS numPins ;
  [- pinName + NET netName
    [+ SPECIAL]
    [+ DIRECTION {INPUT | OUTPUT | INOUT | FEEDTHRU}]
    ...
    [[+ PORT]
      [+ LAYER layerName
        [MASK maskNum]
        [PROPERTY propName propVal] ...
          pt pt
      |+ POLYGON layerName
        [MASK maskNum]
        [PROPERTY propName propVal] ...
          pt pt pt ...
      |+ VIA viaName
        [MASK viaMaskNum]
        [PROPERTY propName propVal] ...
        [orient] pt
      ] ...
      [+ COVER | FIXED | PLACED]
    ] ...
  ; ] ...
END PINS]
```

Example of a DEF 5.8 PINS statement using relative coordinates:

```
PINS 11 ;
 - P0 + NET N0
   + PORT
     + LAYER M2 ( 0 0 ) ( 30 135 )   #This shape is relative to the FIXED value below
     + VIA VIAGEN12_0 ( 0 100 )      #This shape is relative to the FIXED value below
     + FIXED ( 1000 1000 ) N         #Shift the shapes by 1000 1000. The N means no
                                      rotation or flipping
   + PORT
     + LAYER M1 ( 0 0 ) ( 30 135 )   #This shape is relative to the COVER value below
     + VIA VIAGEN12_1 ( 100 0 )      #This shape is relative to the COVER value below
     + COVER ( 1000 2000 ) W         #Shift the shapes by 1000 2000. The W means
                                      rotate 90 degrees
   + PORT
     + POLYGON M2 ( 0 0 ) ( 0 100 ) ( 50 100 ) ( 50 50 ) ( 100 50 ) ( 100 0 )  #This
                                      shape is relative to the PLACED value below
     + PLACED ( 1000 3000 ) S ;      #Shift the shapes by 1000 3000. The S means
                                      rotate 180 degrees
...
```

Example of a DEF 6.0 PINS statement for the same PIN as above:

```
PINS 11 ;
 - P0 + NET N0
   + PORT
     + LAYER M2 ( 1000 1000 ) ( 1030 10135 )
     + VIA VIAGEN12_3  N ( 1000 1100 )
     + FIXED ;
   + PORT
     + LAYER M1 ( 865 2000) ( 1000 2030 )
     + VIA VIAGEN12_1 W ( 1000 2100 )
     + COVER
   + PORT
     + POLYGON M2 ( 1000 3000 ) ( 1000 2900 ) ( 950 2900 ) ( 950 2950 )
       ( 900 2950 ) ( 900 3000 )
     + PLACED ;
...
```

Enhancements in the NETS Regular Wiring Statement

The following enhancements have been made to the NETS Regular Wiring statement:

■ The `PROPERTY` keyword has been added to every shape and via. This enables you to mark specific wires and vias in the NETS routing data, when required.

■ The `SHIELD` keyword has been added to enable you to identify PG net routes that are being used to shield another signal net. This makes it easier to remove shield nets, when required, without disturbing the PG network because the shield nets can be easily identified and differentiated from the PG net routes that are carrying power to PG pins and are therefore part of the PG network.

■ The `NOSHIELD`, `TAPER`, `TAPERRULE`, and `STYLE` routing keywords are considered obsolete and have been removed. See the Obsolete DEF Syntax section for details.

The updated regular wiring syntax with the additions in bold is as follows:

```
{+ COVER | + FIXED | + ROUTED }
    layerName
      [SHIELD]
      [PROPERTY propName propVal] ...
      routingPoints
    [NEW layerName
      [SHIELD]
      [PROPERTY propName propVal] ...
      routingPoints
    ] ...
```

New WIDTH Attribute in the NETS routingPoints Syntax

In previous DEF versions, the routing width was embedded in the `DEFAULT` routing rule or the `NONDEFAULTRULE` (NDR) used for `NETS` routing. As electro-migration (EM), wrong-way routing, and double-patterning rules grew more complex, routers needed to change the width of a net's routing dynamically. This meant artificial NDRs needed to be created.

In DEF 6.0, the `WIDTH` attribute can be defined directly for every routing segment in the `NETS` section. The `DEFAULT` and NDR rules serve as net constraints that routers can override locally to meet local EM, wrong-way, or double-patterning rules. Specifically, if the width of a given routing segment does not match the NET routing-rule width, either `DEFAULT` or `NONDEFAULTRULE`, then it is explicitly given with the route segment using the `WIDTH` keyword. In other words, only the width of the `DEFAULT` or `NONDEFAULTRULE` rule is used by default; any wrong-way route segments that have wrong-way widths require an explicit `WIDTH` value.

Note: The addition of `WIDTH` to NETS routing means the `TAPER` and `TAPERRULE` keywords are no longer needed. Therefore, these are now obsolete as covered in the Obsolete DEF

<u>Syntax</u> section. In addition, the wrong-way routing rule width tables derived from the LEF `WIDTH`, `SPANLENGTHTABLE`, and `WIDTHTABLE` statements are also obsolete.

The updated `NETS` *routingPoints* syntax with the additions in bold is as follows:

```
{ ( x y [extValue] )
    {[MASK maskNum] [WIDTH width] ( x y [extValue] )
    |[MASK viaMaskNum] viaName [orient]
    |[MASK maskNum] RECT ( deltax1 deltay1      deltax2 deltay2 )
    | VIRTUAL ( x y ) } } ...
```

Enhancements in the SPECIALNETS Special Wiring Statement

The following enhancements have been made to the `SPECIALNETS` Special Wiring statement:

- The `PROPERTY` keyword has been added to enable you to mark specific wires and vias in the SPECIALNETS routing data, when required.

- The `+SHIELD` value has been made independent of the of the `COVER`, `FIXED`, `ROUTED` status.

- The `+STYLE` statements are obsolete and have been removed. See the <u>Obsolete DEF Syntax</u> section for details.

The updated special wiring syntax is as follows:

```
[ [+ COVER | + FIXED | + ROUTED]  #+SHIELD removed from here
    [+ SHIELD shieldNetName]
    [+ PROPERTY propName propVal] ...
    [+ SHAPE shapeType] [+ MASK maskNum]
    {+ POLYGON layerName pt pt pt...
    | + RECT layerName pt pt
    | + VIA viaName [orient] pt ...} ...
| {+ COVER | + FIXED | + ROUTED}
    layerName routeWidth
        [+ SHIELD shieldNetName]]
        [+ PROPERTY propName propVal] ...
        [+ SHAPE shapeType]
        routingPoints
    [NEW layerName routeWidth
        [+ SHIELD shieldNetName]]
        [+ PROPERTY propName propVal] ...
        [+ SHAPE shapeType]
```

```
        routingPoints
        ] ...
] ...
```

New PROPERTY and VIA orient Keywords in the FILLS Section

In DEF 6.0, a FILL shape can now have one or more PROPERTY values. The PROPERTY must be defined in the PROPERTYDEFINITIONS section as a SPECIALROUTE property.

In addition, a FILL VIA can now have an *orient* value.

The updated FILLS syntax with the additions in bold is as follows:

```
[FILLS numFills ;
    [- LAYER layerName
        [+ PROPERTY propName propVal] ...
        [+ MASK maskNum] [+ OPC]
            {RECT pt pt | POLYGON pt pt pt ...} ... ;] ...
    [- VIA viaName
        [+ PROPERTY propName propVal] ...
        [+ MASK viaMaskNum] [+ OPC]
            [orient] pt  [[orient] pt] ... ;] ...
END FILLS]
```

Enhancements in the SCANCHAINS Section

The following enhancements have been made in the SCANCHAINS section:

■   A scan chain can now have one or more PROPERTY values. The PROPERTY must be defined in the PROPERTYDEFINITIONS section as a SCANCHAIN property.

■   An ORDERED statement now requires a name. The names might be artificial, such as ordered1, ordered2, and so on. This makes it easier to debug problems between ordered segments of one scan chain. The names do not have to be unique. Multiple ORDERED statements can have the same name.

■   An ORDERED statement can also have its own PROPERTY. The PROPERTY must be defined in the PROPERTYDEFINITIONS section as a SCANCHAIN property.

The updated SCANCHAINS syntax, with the additions in bold, is as follows:

```
[SCANCHAINS numScanChains ;
    [- chainName
        [+ PROPERTY propName propVal] ...
```

```
        [+ PARTITION partitionName [MAXBITS maxbits]]
        [+ COMMONSCANPINS [ ( IN pin )] [( OUT pin ) ] ]
        + START {fixedInComp | PIN} [outPin]
        [+ FLOATING
            {floatingComp [ ( IN pin ) ] [ ( OUT pin ) ] [ ( BITS numBits ) ]} ...]
        [+ ORDERED
            [+ NAME orderName]
            [+ PROPERTY propName propVal] ...
            {fixedComp [ ( IN pin ) ] [ ( OUT pin ) ] [ ( BITS numBits ) ]} ...
        ] ...
         + STOP {fixedOutComp | PIN} [inPin] ]
    ;] ...
END SCANCHAINS]
```

New PROPERTY Keyword in the VIAS Statement

A DEF VIA definition can now have a PROPERTY value attached.

The updated VIAS syntax, with the addition in bold, is as follows:

```
[VIAS numVias ;
    [- viaName
        [+ PROPERTY propName propVal] ...
        [    + VIARULE viaRuleName
                    + CUTSIZE xSize ySize
                    + LAYERS botmetalLayer cutLayer topMetalLayer
                    + CUTSPACING xCutSpacing yCutSpacing
                    + ENCLOSURE xBotEnc yBotEnc xTopEnc yTopEnc
                    [+ ROWCOL numCutRows NumCutCols]
                    [+ ORIGIN xOffset yOffset]
                    [+ OFFSET xBotOffset yBotOffset xTopOffset yTopOffset]
                    [+ PATTERN cutPattern] ]
        | [ + RECT layerName [+ MASK maskNum] pt pt
                    | + POLYGON layerName [+ MASK maskNum] pt pt pt] ...]
    ;] ...
END VIAS]
```

New PROPERTY, WIDTH, and NDR Keywords in TRACKS

The WIDTH and NDR keywords have been added to the TRACKS statement for existing usage. The PROPERTY keyword has also been added to allow for future extensions.

The updated TRACKS syntax, with the additions in bold, is as follows:

```
[TRACKS
    [{X | Y} start DO numtracks STEP space
        [PROPERTY propName propVal] ...
        [MASK maskNum [SAMEMASK]]
        [WIDTH width]
        [NDR ruleName]
        [LAYER layerName ...]
        }...
    ;] ...]
```

Obsolete DEF Syntax

SOURCE, WEIGHT, and REGION Removed from COMPONENTS

The statements and keywords in red are obsolete and have been removed from the
`COMPONENTS` section:

```
COMPONENTS numComps ;
[- compName modelName
    [+ EEQMASTER macroName]
    [+ SOURCE {NETLIST | DIST | USER | TIMING}]
    [+ UNPLACED | {FIXED | COVER | PLACED} pt orient]
    [+ MASKSHIFT shiftLayerMasks]
    [+ HALO [SOFT] left bottom right top]
    [+ ROUTEHALO haloDist minLayer maxLayer]
    [+ WEIGHT weight]
    [+ REGION regionName]
    [+ PROPERTY {propName propVal} ...]...
;] ...
END COMPONENTS
```

■   The `+ SOURCE DIST` value is used to track physical-only cells, that is cells that are not
    in the Verilog netlist. It has been replaced with the new `+ PHYSICAL` keyword (see
    SOFTFIXED, PHYSICAL, and PINPROPERTY Added to COMPONENTS for details).
    The `+ SOURCE NETLIST`, `USER`, and `TIMING` values are obsolete and no longer used
    by any known application, so the complete `+ SOURCE` statement is now obsolete.

■   `+ WEIGHT` is obsolete. It has been replaced with the `SOFTFIXED` placement status (see
    SOFTFIXED, PHYSICAL, and PINPROPERTY Added to COMPONENTS for details).

■   `+ REGION` is obsolete. In previous DEF releases, a `GROUP` was required in order to add
    a `REGION` constraint. This implicitly caused a `GROUP` with the same name as the `REGION`
    to be created, and all components with `+ REGION R1` would be put into a `GROUP` named
    `R1` that had `+ REGION R1` attached. In DEF 6.0, instead of using a `COMPONENT +`
    `REGION` statement, a `GROUP` with `+ COMPONENTS` should be used to list every
    `COMPONENT` in the group and the `GROUP` has a `+ REGION` statement (see New Attributes
    in the GROUPS Section for details).

Multiple Statements and Keywords Removed from the NETS Section

The statements and keywords in red are obsolete and have been removed from the `NETS`
section:

```
NETS numNets ;
    [- { netName
            [ ( {compName pinName | PIN pinName} [+ SYNTHESIZED] ) ] ...
        | MUSTJOIN ( compName pinName ) }
        [+ SHIELDNET shieldNetName ] ...
        [+ VPIN vpinName [LAYER layerName] pt pt
            [PLACED pt orient | FIXED pt orient | COVER pt orient] ] ...
        [+ SUBNET subnetName
            [ ( {compName pinName | PIN pinName | VPIN vpinName} ) ] ...
            [NONDEFAULTRULE rulename]
            [regularWiring] ...] ...
        [+ XTALK class]
        [+ NONDEFAULTRULE ruleName]
        [regularWiring] ...
        [+ SOURCE {DIST | NETLIST | TEST | TIMING | USER}]
        [+ FIXEDBUMP]
        [+ FREQUENCY frequency]
        [+ ORIGINAL netName]
        [+ USE {ANALOG | CLOCK | GROUND | POWER | RESET | SCAN | SIGNAL
                | TIEOFF}]
        [+ PATTERN {BALANCED | STEINER | TRUNK | WIREDLOGIC}]
        [+ ESTCAP wireCapacitance]
        [+ WEIGHT weight]
        [+ PROPERTY {propName propVal} ...] ...
    ;] ...
END NETS
```

- The SYNTHESIZED, VPIN, SUBNET, XTALK, ESTCAP statements have not been in use for many years and are obsolete. They have been removed.

- The + ORIGINAL statement is obsolete and has been removed. It is not used any longer by applications.

- The + SOURCE statement is obsolete. Therefore, [+ SOURCE {DIST | NETLIST | TEST | TIMING | USER}] is obsolete for NETS usage.

- The PATTERN values of WIREDLOGIC and BALANCED have not been supported for many years and are obsolete.

Multiple Routing Keywords Removed from NETS Regular Wiring

The keywords in red are obsolete and have been removed from the NETS regularWiring statement:

```
{+ COVER | + FIXED | + ROUTED | + NOSHIELD}
    layerName [TAPER | TAPERRULE ruleName] [STYLE styleNum]
      routingPoints
    [NEW layerName [TAPER | TAPERRULE ruleName] [STYLE styleNum]
      routingPoints
    ] ...
```

- With the addition of WIDTH to NETS routing in DEF 6.0, any wire-width change from the current net routing rule is given directly, rather than indirectly through TAPER to the

DEFAULT rule width or TAPERRULE to some NONDEFAULTRULE width. Therefore, TAPER and TAPERRULE are no longer required.

- The +NOSHIELD attribute was intended to track the parts of a route had no shield, for example in routing near a pin. However, this attribute has not been in use and current checkers or tools do not look at this data. The attribute is therefore considered obsolete, and has been removed.

- The STYLE *styleNum* was added for X-routing, which is now obsolete. Therefore, it is removed.

NETEXPR Statement Removed from PINS

The NETEXPR statement in the DEF PINS syntax was originally intended to represent OpenAccess power net inheritance rules. However, it is no longer in use and the current practice is to use IEEE1801 or direct global_net_connect statements in its place in digital designs.

Therefore, the NETEXPR statement is considered obsolete and has been removed.

The removed statement in shown in red is below:

```
[PINS numPins ;
    [ – pinName + NET netName
        [+ SPECIAL]
        [+ DIRECTION {INPUT | OUTPUT | INOUT | FEEDTHRU}]
        [+ NETEXPR "netExprPropName defaultNetName"]
        [+ SUPPLYSENSITIVITY powerPinName]
        [+ GROUNDSENSITIVITY groundPinName]
        ...
```

PINPROPERTIES Section Removed

The PINPROPERTIES section has been replaced with DEF PIN + PROPERTY and COMPONENT + PINPROPERTY statements in DEF 6.0.

Therefore, the PINPROPERTIES section shown in red below has been removed in DEF 6.0:

```
[PINPROPERTIES num;
    [- {compName pinName | PIN pinName}
        [+ PROPERTY {propName propVal} ...] ...
    ;] ...
END PINPROPERTIES]
```

STYLES Section Removed

The STYLES section was added for X-routing, which has a wide variety of 45-degree route segments with different types of octagonal end points. X-routing is now obsolete.

Therefore, the STYLES section shown in red below has been removed in DEF 6.0:

```
[STYLES numStyles ;
    {- STYLE styleNum pt pt ... ;} ...
END STYLES]
```

Note: The implicit default style sometimes used by 45-degree route-segments for bump routing is still supported. See the "Default Style for 45-Degree Routing" section in the DEF Syntax chapter of the *LEF/DEF Language Reference* for details on the default style.

Any usage of explicitly declared STYLES is obsolete. Instead, use a POLYGON in the SPECIALNETS wiring statement.

Multiple Statements and Keywords Removed from SPECIALNETS

The statements and keywords in red are obsolete and have been removed from the SPECIALNETS section:

```
[SPECIALNETS numNets ;
    [- netName
        [ ( {compName pinName | PIN pinName} [+ SYNTHESIZED] ) ] ...
     [+ VOLTAGE volts]
     [specialWiring] ...
     [+ SOURCE {DIST | NETLIST | TIMING | USER}]
     [+ FIXEDBUMP]
     [+ ORIGINAL netName]
     [+ USE {ANALOG | CLOCK | GROUND | POWER | RESET | SCAN | SIGNAL | TIEOFF}]
     [+ PATTERN {BALANCED | STEINER | TRUNK | WIREDLOGIC}]
     [+ ESTCAP wireCapacitance]
     [+ WEIGHT weight]
     [+ PROPERTY {propName propVal} ...] ...
    ;] ...
END SPECIALNETS]
```

These keywords have been removed to keep the SPECIALNETS section in sync with the NETS section. For an explanations of these changes, see Multiple Statements and Keywords Removed from the NETS Section.

STYLE and BLOCKAGEWIRE Removed in SPECIALNETS Special Wiring

The keywords in red are obsolete and have been removed from the SPECIALNETS Special Wiring statement:

```
    [[+ COVER | + FIXED | + ROUTED | + SHIELD shieldNetName]
        [+ SHAPE
                {RING | PADRING | BLOCKRING | STRIPE | FOLLOWPIN
                | IOWIRE | COREWIRE | BLOCKWIRE | BLOCKAGEWIRE | FILLWIRE
                | FILLWIREOPC | DRCFILL}]
        [+ MASK maskNum]
        {+ POLYGON layerName pt pt pt ...
        | + RECT layerName pt pt
        | + VIA viaName [orient] pt ...}...
    |{+ COVER | + FIXED | + ROUTED | + SHIELD shieldNetName}
        layerName  routeWidth
            [+ SHAPE
                {RING | PADRING | BLOCKRING | STRIPE | FOLLOWPIN
                | IOWIRE | COREWIRE | BLOCKWIRE | BLOCKAGEWIRE | FILLWIRE
                | FILLWIREOPC | DRCFILL}]
            [+ STYLE styleNum]
            routingPoints
        [NEW layerName routeWidth
            [+ SHAPE
                {RING | PADRING | BLOCKRING | STRIPE | FOLLOWPIN
                | IOWIRE | COREWIRE | BLOCKWIRE | BLOCKAGEWIRE | FILLWIRE
                | FILLWIREOPC | DRCFILL}]
            [+ STYLE styleNum]
            routingPoints
        ] ...
] ...
```

■ The BLOCKAGEWIRE shape was used in earlier DEF versions when a routing BLOCKAGE did not exist. It is now obsolete, and no longer needed. A routing BLOCKAGE should be used instead.

■ The + STYLE statement in the special wiring statement was used for X-routing, which is now considered obsolete. The + STYLE statement has therefore been removed.

Note: The implicit default style sometimes used by 45-degree route-segments for bump routing is still supported. See the "Default Style for 45-Degree Routing" section in the DEF Syntax chapter of the *LEF/DEF Language Reference* for details on the default style.

Any usage of explicitly declared STYLES is obsolete. Instead, use a POLYGON in the SPECIALNETS wiring statement.

SLOTS Section Removed

Slots in metal were intended to reduce metal density. Although some older technologies still use slots, but DEF support was never used for them. Therefore, the following syntax shown in red is obsolete and has been removed:

■ SLOTS section

```
[SLOTS numSlots ;
    [- LAYER layerName
```

```
        {RECT pt pt | POLYGON pt pt pt ... } ...
    ;] ...
    END SLOTS]
```

■   SLOTS keyword in the BLOCKAGES section

```
[BLOCKAGES numBlockages ;
    [- LAYER layerName
        [ + COMPONENT compName | + SLOTS | + FILLS | + PUSHDOWN
            | + EXCEPTPGNET]
        ...
```

DESIGNRULEWIDTH Removed from DEF BLOCKAGES

The DESIGNRULEWIDTH attribute was intended to model abstract blockages using spacing rules for a given design rule width. However, it is not in use and has been removed from the BLOCKAGES syntax, as indicated in red below.

```
[BLOCKAGES numBlockages ;
    [- LAYER layerName
      [+ COMPONENT compName | + FILLS | + PUSHDOWN
        | + EXCEPTPGNET]
      [+ SPACING minSpacing | + DESIGNRULEWIDTH effectiveWidth]
        {RECT pt pt | POLYGON pt pt pt ...} ...
    ;] ...
```

Note: LEF OBS DESIGNRULEWIDTH is still in use and has not been made obsolete.

EXTENSIONS Section Removed

The EXTENSIONS section was an alternative to the PROPERTY usage to add customized data to a DEF file. However, it has not been in use as all tools use a string PROPERTY instead to pass additional or custom data.Therefore the EXTENSIONS section shown below in red is obsolete and has been removed:

```
[BEGINEXT "tag"
    extensionText
ENDEXT]
```

3

ALIAS Statement

ALIAS Statement Removed

The `ALIAS` statement has not been in use for a long time now. It was originally added for gate-arrays to encapsulate some idioms for power routing tied to a particular gate array. However, in current technologies, this is done through Tcl scripts. Therefore, the usage shown below in red is obsolete:

```
&ALIAS  &&aliasName  =  aliasDefinition   &ENDALIA
```

Note: Aliases processing will be disabled not only in LEF/DEF version 6.0 files but also for all older LEF/DEF version files processing. If any of your old files still use ALIAS statements, read them into your tool's last version that supports ALIAS (for Innovus, the last version supporting ALIAS is Innovus 20.1x), and then write the file back out to remove the ALIAS usage.