QUALiS DESIGN CORPORATION

VHDL QUICK REFERENCE CARD

REVISION 1.1

()	Grouping	[ ]	Optional
{}	Repeated	|	Alternative
bold	As is	CAPS	User Identifier
italic	VHDL-1993		

1. LIBRARY UNITS

[{use_clause}]
entity ID **is**
  [**generic** ({ID **:** TYPEID [**:=** expr]**;**})**;**]
  [**port** ({ID **:** **in** | **out** | **inout** TYPEID [**:=** expr]**;**})**;**]
  [{declaration}]
[**begin**
  {parallel_statement}]
end [***entity***] ENTITYID**;**

[{use_clause}]
architecture ID **of** ENTITYID **is**
  [{declaration}]
begin
  [{parallel_statement}]
end [***architecture***] ARCHID**;**

[{use_clause}]
package ID **is**
  [{declaration}]
end [***package***] PACKID**;**

[{use_clause}]
package body ID **is**
  [{declaration}]
end [***package body***] PACKID**;**

[{use_clause}]
configuration ID **of** ENTITYID **is**
for ARCHID
  [{block_config | comp_config}]
end for;
end [***configuration***] CONFID**;**

use_clause::=
  library ID**;**
  [{**use** LIBID**.**PKGID**.all;**}]

block_config::=
  for LABELID
    [{block_config | comp_config}]
  end for;

comp_config::=
  for all | LABELID **:** COMPID
    (**use entity** [LIBID**.**]ENTITYID [**(** ARCHID **)**]
      [[**generic map (** {GENID **=>** expr **,**} **)**]
      port map ({PORTID **=>** SIGID | *expr* **,**}**)**]**;**
    [**for** ARCHID
      [{block_config | comp_config}]
    end for;]
    end for;) |
    (**use configuration** [LIBID**.**]CONFID
      [[**generic map (**{GENID **=>** expr **,**}**)**]
      port map ({PORTID **=>** SIGID | *expr***,**}**)**]**;**)
  end for;

2. DECLARATIONS

2.1. TYPE DECLARATIONS

type ID **is (** {ID**,**} **);**

type ID **is range** number **downto** | **to** number**;**

type ID **is array (** {range | TYPEID **,**}**)**
  of TYPEID | SUBTYPID**;**

type ID **is record**
  {ID **:** TYPEID**;**}
end record;

type ID **is access** TYPEID**;**

type ID **is file of** TYPEID**;**

subtype ID **is** SCALARTYPID **range** range**;**

subtype ID **is** ARRAYTYPID**(** {range**,**}**)**

subtype ID **is** RESOLVFCTID TYPEID**;**

range ::=
  (integer | ENUMID **to** | **downto**
  integer | ENUMID) | (OBJID**'**[**reverse_**]**range**) |
  (TYPEID **range <>**)

2.2. OTHER DECLARATIONS

constant ID **:** TYPEID **:=** expr**;**

[***shared***] **variable** ID **:** TYPEID [**:=** expr]**;**

signal ID **:** TYPEID [**:=** expr]**;**

file ID **:** TYPEID (**is in** | **out** string**;**) |
  (*open read_mode* | ***write_mode***
  | ***append_mode is*** *string;***)**

alias ID **:** TYPEID **is** OBJID**;**

attribute ID **:** TYPEID**;**

attribute ATTRID **of** OBJID | **others** | **all :** class
  is expr**;**

class ::=
  entity | **architecture** | **configuration** |
  procedure | **function** | **package** | **type** |
  subtype | **constant** | **signal** | **variable** |
  component | **label**

component ID [***is***]
  [**generic (** {ID **:** TYPEID [**:=** expr]**;**} **);**]
  [**port (**{ID **: in** | **out** | **inout** TYPEID [**:=** expr]**;**}**);**
end component [*COMPID*]**;**

[***impure***] **function** ID
  [**(** {[**constant** | **variable** | **signal**] ID **:**
    in | **out** | **inout** TYPEID [**:=** expr]**;**}**)**]
  return TYPEID [**is**
begin
  {sequential_statement}
end [***function***] ID]**;**

procedure ID[**(**{[**constant** | **variable** | **signal**] ID **:**
    in | **out** | **inout** TYPEID [**:=** expr]**;**}**)**]
[**is begin**
  [{sequential_statement}]
end [***procedure***] ID]**;**

for LABELID | **others** | **all :** COMPID **use**
  (**entity** [LIBID**.**]ENTITYID [**(** ARCHID **)**]) |
  (**configuration** [LIBID**.**]CONFID)
    [[**generic map (** {GENID **=>** expr**,**} **)**]
    port map ({PORTID **=>** SIGID | *expr***,**} **)**]**;**

3. EXPRESSIONS

expression ::=
  (relation **and** relation) |
  (relation **or** relation) |
  (relation **xor** relation)

relation ::=	shexpr [relop shexpr]
shexpr ::=	sexpr [*shop sexpr*]
sexpr ::=	[**+**|**-**] term {addop term}
term ::=	factor {mulop factor}

factor ::=
  (prim [****** prim]) | (**abs** prim) | (**not** prim)

prim ::=
  literal | OBJID | OBJID**'**ATTRID | OBJID**(**{expr**,**}**)**
  | OBJID**(**range**)** | **(**{[choice [{**|** choice}] **=>**] expr**,**}**)**
  | FCTID**(**{[PARID **=>**] expr**,**}**)** | TYPEID**'(**expr**)** |
  TYPEID**(**expr**)** | **new** TYPEID[**'(**expr**)**] | **(** expr **)**

choice ::=	sexpr | range | RECFID | **others**

3.1. OPERATORS, INCREASING PRECEDENCE

logop	**and** | **or** | **xor**
relop	**=** | **/=** | **<** | **<=** | **>** | **=>**
shop	***sll*** | ***srl*** | ***sla*** | ***sra*** | ***rol*** | ***ror***
addop	**+** | **-** | **&**
mulop	***** | **/** | **mod** | **rem**
miscop	****** | **abs** | **not**