







File does not exist.

Und was, wenn doch? Wahrscheinlich heißt dein Projektordner “Übung...”.

“Ü”, “Ö”, “Ä” und ihre kleinen Brüder sowie das “ß” sind verboten.

Niemals irgendwo in Diamond verwenden. Auch nicht in Kommentaren. Nichts

ist sicher vor den Umlauten. Des Weiteren ist das Beginnen von Dateinamen

mit Zahlen oder Sonderzeichen auch nicht schlau. Oder das Verwenden von

## Schlüsselworten in Dateinamen und Variablen.



-G-



-H-

[illegible]

-I-

indexed name prefix is not an array type. (VDHL-1122)

Ein Vektor bzw. Array wurde erwartet, aber die Variable ist keiner.

Tritt u.a. bei einem falschen Typecast mit der

to\_integer-Funktion auf.

### Beispiele:

\*1\*

```
ceh: out unsigned(3 downto 0);
```

type int32 is range 0 to 31;

```
signal deh: int32;
```

```
???    ceh <= to_unsigned(to_integer(deh), 4);
```

```
!!!    ceh <= to_unsigned(integer(deh), 4);
```

[illegible]

-J-



-K-

[illegible]

-L-

ERROR - logical block <blockname> with type <entityname> is unexpanded.

Done: error code 2

Dieser Fehler tritt auf, wenn ein Block, der im Schematic eingesetzt

wurde, nachträglich so verändert wurde, dass er keine Outputs mehr hat.

Beispiel:

\*1\*

entity <entityname> is

```
port(    ah: in std_logic;
```

```
    beh: out std_logic);
```

end entity;

architecture <archname> of <entityname> is

```

signal beh_sig: std_logic;

begin
process(ah) is
if rising_edge(ah) then
    beh_sig <= not beh_sig;
end if;
end process;

!!!    beh <= beh_sig;

end architecture;

-- wenn man sich jetzt die Zuweisung von beh_sig nach beh
-- entfernt, ist beh nicht mehr intern verbunden und wird wegoptimiert.
-- der Block im Schematic ist jetzt "unexpanded".

entity <entityname> is
port(  ah: in std_logic;
       beh: inout std_logic);
end entity;

architecture <archname> of <entityname> is

signal beh_sig: std_logic;

begin
process(ah) is
if rising_edge(ah) then
    beh_sig <= not beh_sig;
end if;
end process;

???    -- wir haben die Zuweisung von beh_sig nach beh entfernt.

end architecture;

```

---

## Linux

Es existiert eine Linuxversion von Diamond für Distributionen, die .rpm-Pakete verwenden. Ich habe diese Version auf Fedora 28 getestet. Was funktioniert: Code schreiben und Syntax korrigieren lassen, Synthesis (wenn der Code fehlerfrei ist). Was nicht funktioniert: Der Programmierer, das HDL-Diagramm. Für den Programmierer braucht man Treiber, die nicht so ohne Weiteres zu finden sind. Zudem stürzt Diamond in unregelmäßigen Abständen nicht-reproduzierbar ab: es ist nicht sehr stabil. Ich rate von der Verwendung von Diamond unter Linux ab und rate zur Installation von Windows 7 X64 auf einer eigenen Partition. 30GB sollten für das

Betriebssystem und Diamond reichen (Diamond braucht ca. 8GB, Windows 7 mit Updates ca. 20GB), aber ich empfehle, die Partition etwas großzügiger zu wählen, um im späteren Studienverlauf z.B. Matlab und LTSpice installieren zu können.

-M-

Dieser Fehler tritt unter anderem auf, wenn im Schematic ein Netz sowohl als Input festgelegt wurde, als auch von vhi/vlo beschaltet wird.

-N-

Eine Variable / ein Signal / ein Output wird von mehreren Quellen beschaltet.

\*1\*



-0-

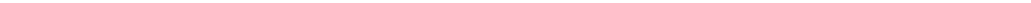
-P-

Rudelfehler, tritt oft in Massen auf. Er kann verschiedene Ursachen haben. Zum Beispiel wurden seit der letzten Verschaltung im Spreadsheet View Pins entfernt, die belegt waren. Zum Beheben dieses Fehlers im Reiter “Files” unter der Rubrik “LPF Constraint Files” die

Ein anderer Fall ist, wenn Ports wegoptimiert werden. Meist ist das darauf zurückzuführen, dass die Ports wirklich nicht benötigt werden. Es wird ein Signal nicht in den Output geschrieben, oder der Input nicht in ein Signal, oder die In- und Outputs in Frage tauchen generell nicht im Code auf.

The programming file doesn't match the file generated by the project.  
JEDEC file <path\to\file>  
Check configuration setup: Unsuccessful.

Q-

  
-R-

[illegible]

NOT(clock-edge) condition. (VHDL-1242)

Ein Problem, das beim Arbeiten mit Clocks auftritt. Das Grundproblem ist das mögliche Zugreifen auf ein Netz (Variable, Signal, I/O...) von mehreren Quellen.

## Beispiele:

```
*1*
??? ah <= '0';
    if rising_edge(clk) then
???     ah <= '1';
```

-- Eins der Statements muss  
-- entfernt werden, damit  
-- es wieder funktioniert.

```

*2*
if rising_edge(clk1) then
???    ah <= '1';
end if;
if rising_edge(clk2) then
???    ah <= '0';           -- Wie *1*.
end if;

```

```

*3*
??? if rising_ege(clk) or
??? falling_edge(clk) then      -- Wie *1*.
    end if;

```

[illegible]

-T-

type error near <this> ; expected type <sometype> (VHDL-1272)

Was auf der rechten und linken Seite der Zuweisung steht, ist nicht der gleiche Datentyp (\*1\*). Außerdem kann es sein, dass ein Typecast falsch angewendet wurde (\*2\*).

Beispiele:

```

*1*
Ah: in std_logic_vector(3 downto 0);
Ceh: out unsigned(3 downto 0);
??? Ceh <= Ah;
!!! Ceh <= unsigned(Ah);

```

```
*2*
ceh: out unsigned(3 downto 0);
type int32 is range 0 to 31;
signal deh: int32;
??? ceh <= to_unsigned(deh, 4);
!!! Ceh <= to_unsigned(integer(deh), 4);
```



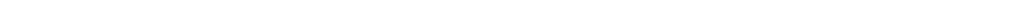
-U-



## Unexpected EOF (VHDL-1283)

Syntaxfehler, entity oder architecture wurden nicht korrekt geschrieben (die Blöcke, nicht die Worte). Ich rate, zu überprüfen, ob alle “begin”, “end architecture”, “port(, “); end entity” eingehalten wurden.

[illegible]



-W-





-Y-



0 definitions of operator "<operator>" match here (VHDL-1052)

Der Operator ist nicht für diese Datentypen definiert. Es fehlt entweder die richtige Bibliothek (use `ieee.numeric_std.all`; use `ieee.std_logic_1164.all`); oder die Datentypen vertragen sich nicht mit dem Operator, siehe “DL\_Anleitung\_XO2\_180417.pdf”, Seite 134. Oder aber die Datentypen stimmen nicht überein. In diesem Fall hilft ein `typecast` mit `to_unsigned()`, `unsigned()`, `integer()`, `to_integer()`, `std_logic_vector()`, je nach Situation.

Beispiele:

\*1\*

Ah: in std\_logic\_vector(3 downto 0);

```
Beh: in unsigned(3 downto 0);
```

```
Ceh: out unsigned(3 downto 0);
```

???    Ceh <= Ah + Beh;

```
!!!    Ceh <= unsigned(Ah) + Beh;
```

[illegible]

Anderes

Schematic setzt nur die alten .sym-Dateien bzw Blöcke ein.

Ein Problem, das beim Arbeiten mit Diamond auftritt. Zuerst den Tab mit der Schematic schließen und wieder öffnen. (Wenn man Tabs von Programmen in Diamond schließt, verliert man auch die Strg-Z-Historie.) Wenn das Problem besteht, die Schematic-File von der Implementierung entfernen (Rechtsklick – Exclude from Implementation) und wieder einbinden (Rechtsklick – Include in Implementation) und mit einem Doppelklick die .sch-File wieder öffnen.

Alternativ: Diamond schließen und neu starten.

Nach einer Archivierung des Projekts kann es zu seltsamen Fehlern kommen.

Lösung: Diamond schließen und neu öffnen. Das Projekt kann auch gespeichert werden, das Problem liegt bei Diamond.

---

Die Clock hält an.

Lösung: In der Diamond-Bibliothek für logische Blöcke findet man den Block "GSR". Setzt man ihn high, also '1', läuft die Clock immer. Setzt man ihn low, also '0', läuft die Clock nie.

---

LED(s) leuchten weniger stark als üblich.

Das Problem ist, dass die LED weniger Spannung als üblich erhält. Das kann unterschiedliche Ursachen haben. Eine wäre zum Beispiel unterbrochene Schaltvorgänge. Das heißt, wir sind weder bei '0' noch bei '1', weil der Schaltvorgang von '1' zu '0' oder '0' zu '1' unterbrochen wurde. Das kann durch ungeeignete Realisierung des Codes durch den Compiler (da auf dem PLD parallel gearbeitet wird, wir aber seriellen Code schreiben) passieren. Die Lösung ist, den "gleichen" Code mit anderen Befehlen zu schreiben.

Ein anderes häufiges Problem, welches zu diesem Symptom führt, ist Schwingen. Wenn ein Output schwingt, leuchtet die LED in der Regel zu ca. 50%. Dann wurde vermutlich ein "combinational loop thru inverter" gebaut, der nicht als solcher erkannt wurde. Siehe auch den gleichnamigen Fehler unter dem Buchstaben C.

---

Unterschied Signal und Variable

Eine Variable wird direkt im Prozess übernommen, ein Signal erst, wenn der Prozess anhält, und nur die letzte Veränderung gilt. Outputs verhalten sich ähnlich wie Signale (\*1\*). Zudem sind Variablen nur innerhalb eines Prozesses verfügbar (\*2\*).

Beispiele:

\*1\*:

ah: out std\_logic;

beh: out std\_logic;

```
architecture <archname> of <entityname> is
```

```
    signal my_sig: std_logic := '0';
```

```
begin
```

```
process(my_clk) is
```

```
    variable my_var: std_logic := '0';
```

```
begin
```

```
if rising_edge(my_clk) then
```

```
    my_var := '1';
```

```
    if my_var = '1' then
```

```
        ah <= '1'; -- dies wird direkt getan
```

```
    end if;
```



moeglich '0'.

```
-- my_sig: _____ ... <-    my_sig wird erst  
                                uebernommen, wenn der  
                                Prozess stoppt. Somit  
                                bleibt es '0'.
```

\*2\*

```
ah: in std_logic;  
beh: out std_logic;
```

```
architecture <archname> of <entityname> is  
begin  
process(ah) is
```

```
variable ah_var: std_logic;
```

```
begin
```

```
if rising_edge(var) then  
    ah_var <= not ah_var;  
end if;
```

```
!!!    beh <= ah_var;        -- das geht mit Variablen und Signalen  
  
end process;
```

```
???    beh <= ah_var;        -- das geht nicht mit Variablen, nur mit  
                                -- Signalen  
end architecture;
```

---

Der Synthesis-Prozess endet nicht, er läuft endlos.

Möglicherweise ist im Schematic ein Output als Input gekennzeichnet.

---

Groß- und Kleinschreibung...

... gibt es in VHDL nicht. Demnach sind "Ah", "aH", "AH" und "ah" das Gleiche für Diamond.

---

Andere Dinge, nach denen man Ausschau halten sollte:

Keine Umlaute verwendet?

Nicht aus Versehen ein Schlüsselwort benutzt? Die Vektoren

B[start:finish] und X[start:finish] und einige Weitere können

zu Problemen führen, da sie teilweise schon von Diamond für bestimmte Funktionen genutzt werden. Nicht im Schematic verwenden! Allgemeine

Empfehlung: eindeutige Variablennamen und keine Schlüsselwörter wie integer, null, to, then, ... verwenden. Keine kurzen Namen wie A, B, X, ... wählen. Des Weiteren: "Input" und "Output" sind Schlüsselwörter. Bitte keine Signale, Variablen oder I/O so benennen.