library ieee; use ieee.std logic 1164.all; use ieee.numeric_std.all; use work.all; CATCH UNKNOWN EXCEPTION: DAS # LATTICE (a) # **DIAMOND** # # # **FEHLERLEXIKON** <Ein Projekt von Benedikt Reinberger> Plaintext Version 1.0 Diamond Version 3.10.0.111.2 Windows 7 Pro SP1 x64 Build 7601 Disclaimer: Dieses Dokument ist nur durch den Autor getestet und kann demnach Fehler aufweisen. Die folgenden Zeilen sind ausschließlich als Orientierungshilfen zur Fehlerdiagnose gedacht. Des Weiteren: Das .pdf wird nicht weitergeführt, die Plaintext-File wird in Zukunft die aktuellsten Änderungen beinhalten. Grund dafür ist simplere Ergänzung des Lexikons. Außerdem: Strg+F ist dein Freund. Fragen, Anregungen: Hast du einen Fehler im Dokument gefunden? Ist etwas nicht klar

Hast du einen Fehler im Dokument gefunden? Ist etwas nicht klar verständlich? Bist du einem Fehler begegnet, der noch nicht hier festgehalten ist? Sprich mich im Labor an oder schreibe mir eine Mail an rebe1011@hs-karlsruhe.de

Legende:

??? markiert eine Zeile mit Fehlern,

!!! markiert eine Zeile mit Lösungsvorschlägen.

><><><><><>	><><>
A-	
><><><><>	><><>
B-	
><><><>	><><>
G-	

```
Diamond weiß nicht, wie es den Typ zu konvertieren hat. Man muss also
ein paar weitere Typecasts machen.
Beispiele:
       *1*
       ceh: out unsigned(3 downto 0);
       type int32 is range 0 to 31;
       signal deh: int32;
       ceh <= unsigned(deh);</pre>
???
!!!
       ceh <= to unsigned(integer(deh), 4);
Cannot update 'in' object <inputname> (VHDL-1327)
Man kann einem Input keine Werte zuweisen.
Beispiele:
       *1*
       ah: in std_logic;
???
       ah <= '1';
Combinational loop thru inverter found at net: <netname> driven by
instance: <instance>
Das passiert zum Beispiel, wenn man einen schwingenden Output bauen
möchte, oder versehentlich gebaut hat. Zur Lösung kann man z.B. eine
Clockabhängigkeit ins Spiel bringen – je nach dem, für was der Block
ursprünglich geplant war.
Beispiele:
       *1*
       myclk: in std logic;
       ah: inout std_logic; --Hier: output
       architecture <archname> of <entityname> is
       begin
!!!
       process (myclk) is
!!!
       begin
!!!
              if rising_edge(myclk) then
!!!
                      ah <= not ah;
!!!
              end if:
!!!
       end process;
???
       -- Alles außerhalb eines Prozesses ist sozusagen "Hard-Wired."
       -- Demnach wird ah so schnell negiert, wie
???
???
       -- die Gatter schalten koennen.
???
              ah <= not ah;
       end <archname>;
```

-D-

-E-

Exception Condition: core file <path\to\dump> is generated For support, contact Lattice Semiconductor technical support at 1-800-LATTICE or 503-268-8001, or techsupport@latticesemi.com.

Done: error code 999

Zuerst: Wir rufen nicht Lattice an, die haben doch selbst keinen Schimmer. Des Weiteren: Teile keine Vektoren, die 4 lang sind, durch 10. Sondern erst durch 5 und dann durch 2. Alternativ können wir die Länge des Vektors in Frage verändern und dann Formpressung mit resize(<vector>, <sizeof_vector_in_integer>) anstellen. Beispiele:

```
*1*
        Ah: in unsigned(1 \text{ to } 4);
        Beh: out unsigned(1 to 4);
???
        Beh \leq Ah / 10;
!!!
        Beh \leq Ah / 5 / 2;
        *2*
        Ah: in unsigned(1 \text{ to } 5);
        Beh: out unsigned(1 to 4);
???
        Beh \leq Ah / 10;
                                -- das funktioniert nicht, weil die Groessen
???
                        -- der Vektoren nicht gleich sind -> VHDL-1077
!!!
        Beh \leq resize(Ah / 10, 4);
```

Ein Weiterer Fall, bei dem dieser Fehler auftritt, ist das gleichzeitige Nutzen von vhi und vlo am selben Netz im Schematic.

```
expression has <N> elements; expected <M> (VHDL-1077)
```

Die Größen der Vektoren auf beiden Seiten der Zuweisung stimmen nicht. Abhilfe schafft bei signed und unsigned die resize(<vector>, <sizeof_vector_in_integer>)-Funktion. Alternativ kann man mit Verkettung arbeiten. Beispiele:

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-
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);</pre>
!!!
    ceh <= to_unsigned(integer(deh), 4);</pre>
−J−
```

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

Done: error code 2

-L-

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, Sythesis (wenn der Code fehlerfrei ist). Was nicht funktioniert: Der Programmer, das HDL-Diagramm. Für den Programmer 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.

Multiple Non-Tristate drivers exist for net <netname>.

Done: error code 2

Dieser Fehler tritt unter anderem auf, wenn im Schematic ein Netz sowohl als Input festgelegt wurde, als auch von

vhi/vlo beschaltet wird.

net <netname> is constantly driven from multiple places Done: error code 2

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

```
Beispiele:
```

```
*1*
???
    if beh = '1' then
         ah <= '0':
???
???
    end if:
???
    if ceh = '1' then
???
         ah <= '1';
???
     end if;
    if beh = '1' then
!!!
!!!
         ah <= '0'
!!!
     elsif ceh = '1' then
                    -- Diese Loesung schliesst aus, dass ah
                    -- gleichzeitig beschaltet werden kann.
!!!
!!!
          ah <= '1';
                    -- Allerdings wird jetzt ah nur '1'
                    -- gesetzt, wenn beh '0' ist UND ceh '1'
!!!
!!!
                    -- ist.
!!!
     end if:
-P-
```

The port [<portname>] doesn't exist in the design.

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

f-Datei öffnen. Die Pins, die nicht existieren, löschen,
dann speichern. Nach erneuter Synthesis wird der Fehler nicht mehr
auftauchen.

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.

Der Fehler tritt auch manchmal auf, wenn ein .sym neu generiert wurde. Wenn der Reiter der Schematic-File geschlossen und wieder geöffnet wird, wird der neue Block eingesetzt – und oft passiert es, dass dann Drähte nicht mehr verbunden sind (auch wenn das Layout des Blocks gleich ist!). Zu erkennen ist dieser Fehler an den Vierecken an den Enden von Drähten, die verbunden sein sollten, oder an Drähten, die jetzt in der Luft hängen. Hat sich das Layout des Blocks verändert, einfach die Drähte neu ziehen. Ist das Layout gleich geblieben, den Block ausschneiden und wieder an der gleichen Stelle einfügen.

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

Passiert häufig bei archivierten Projekten. Lösung: Unter "file name" auf den Pfad klicken und entweder manuell ändern oder auf die Schaltfläche mit den drei Punkten klicken und im Explorer zur richtigen Datei navigieren und diese wählen.

<><><><><>-Q-

Qt: Untested Windows Version 6.2 detected!

Done: error code 1

Fehler, der hauptsächlich bei der Verwendung von Windows 10 auftritt. Lösung: Windows-Taste drücken, in das Suchfeld "Diamond" eingeben. Die Höchste Übereinstimmung sollte das Symbol von Diamond mit diesem Namen anbei sein. Rechtsklick auf das Symbol > "Dateispeicherort öffnen". Bei den Rechnern im Poolraum sollte nun in einem Ordner mit ein paar Verknüpfungen landen. Rechtsklick auf die Verknüpfung von Diamond > Eigenschaften > Kompatibilität > Haken setzen bei "Programm im Kompatibilitätsmodus ausführen" > Im Dropdown-Menü "Windows 7" auswählen > Übernehmen > OK.

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.
     end if:
     *2*
     if rising_ege(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:
-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:

```
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.
-V-
-W-
-X-
-Z-
```

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:

Ah: in std_logic_vector(3 downto 0); Beh: in unsigned(3 downto 0): Ceh: out unsigned(3 downto 0); $Ceh \le Ah + Beh$:

???

Ceh <= unsigned(Ah) + Beh;

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:

```
my_sig <= '1';
       if my sig = '1' then
              beh <= '1';
                             -- das wird erst einen
                             -- my_clkzyklus
                             -- spaeter getan.
       end if:
end if;
end process;
end architecture;
-- Signalverlauf (nicht skalentreu):
-- my_clk:
-- my_var:
                  ↓ Hier endet der Prozess.
-- ah:
-- my_sig:
                         ↓ Hier endet der Prozess.
-- beh:
*2*
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'; -- rising edge bei my_var
       my_var := '0'; -- falling edge bei my_var
       my_sig <= '1'; -- my_sig wird am Ende des Prozesses '1'
                     -- zugewiesen
       my_sig <= '0'; -- jetzt wird my_sig am Ende des
                     -- Prozesses '0'
                     -- zugewiesen, die vorherige Zuweisung
                     -- wurde ueberschrieben.
                     -- Es wurden keine edges erzeugt.
end if;
end process;
end architecture;
-- Signalverlauf (nicht skalentreu):
-- my_clk: ____ | ...
               ↓ Hier endet der Prozess.
-- my_var: ____||___||__... <-
                                           my var wird so schnell
                                    wie moeglich '1' und
                                    dann so schnell wie
```

moeglich '0'.

```
my_sig wird erst
       -- my_sig: _____
                                           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.