

Installation und Inbetriebnahme der benötigten Software

Für die Bearbeitung der Aufgabe 1 werden zwei Programme benötigt, wobei es sich um den Simulator GHDL und den Waveformviewer GTKWave handelt.

1. Speichert die Datei *"ILC2010_A1_Software.exe"* auf eurem Computer z.B. in dem Verzeichnis *c:*
2. Öffnet den Windows Explorer und führt die Datei *"ilca1.exe"* durch einen Doppelklick aus. Es handelt sich hierbei um eine selbstentpackende ZIP-Datei.
3. Wählt als Zielverzeichnis wie vorgeschlagen *c:\ilca1* und klickt auf "Unzip" um das Entpacken zu starten.
4. Nach dem Auspacken könnt ihr das ZIP-Programm schließen.
5. Wechselt im Explorer zu dem Verzeichnis *c:\ilca1*
6. Ein Doppelklick auf *"gtkwave.exe"* sollte jetzt den Waveform-Viewer GTKWave öffnen, den ihr benötigt um euch die Simulationsergebnisse aus Aufgabenteil 1.2 und 1.3 anzuschauen.
7. Schließt GTKWave wieder.
8. Startet im Explorer die Datei *"ghdl-installer-0.26.exe"*, um den Simulator GHDL zu installieren.
9. Lest und bestätigt die Lizenz. Es handelt sich um ein freies Programm unter der GPL.
10. Installiert alle bereits ausgewählten Komponenten.
11. Wählt als Installationsverzeichnis *"c:\ilca1"*.
12. Um GHDL zu starten, öffnet eine Eingabeaufforderung, die sich normalerweise unter "Start->Programme->Zubehör->Eingabeaufforderung" befindet.
13. Wechselt in das Verzeichnis *ilca1*, indem ihr den Befehl *„cd c:\ilca1“* eintippt und mit der Enter-Taste bestätigt.
14. Tippt *"ghdl -v"* ein und bestätigt mit der Enter-Taste. Es sollte nun eine Information zur Version von GHDL auf dem Bildschirm erscheinen.
15. Wenn Ihr sowohl GTKWave als auch GHDL starten konntet, ist die Software richtig installiert.

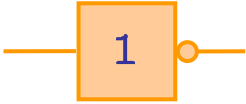


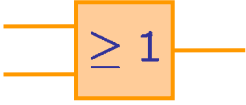
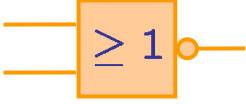
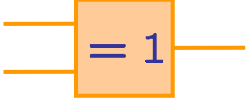
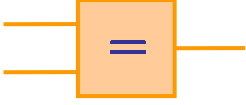
Simulator GHDL

- Wechselt zunächst in das hdl Verzeichnis (*cd c:lilca1\hdl*).
- `ghdl.exe -a full_adder.vhdl` (übersetzt die in der Datei angegebenen Schaltung)
- `ghdl.exe -a full_adder_test.vhdl` (übersetzt die passende Testumgebung)
- `ghdl.exe -r full_adder_test --vcd=full_adder_test.vcd` (startet die Simulation von `full_adder_test` und erzeugt die Waveform `full_adder_test.vcd`. *War die Simulation erfolgreich, erscheint die Ausgabe "end of test" in der Eingabeaufforderung.*)

Waveformviewer GTKWave

- Die durch die Simulation erzeugte Waveform kann mit GTKWave angesehen werden.
- Startet GTKWave. Ihr könnt die Waveform-Datei `full_adder_test.vcd` öffnen, indem ihr den Menüpunkt "File->Open New Tab" wählt, in dem erscheinenden Dateibrowser in das Unterverzeichniss "hdl" wechselt und die Datei auswählt (Alternativ könnt ihr die Datei "full_adder_test.vcd" mit "Drag and Drop" aus Windows Explorer auf das GTKWave-Fenster ziehen.).
- Ihr könnt nun die Signale der Testumgebung (links) in das Fenster „Waves“ (rechts) ziehen. Beobachtet, wie sich die Signalpegel in Abhängigkeit der Zeit ändern.
- Tipps:
 - o Mit der Lupenfunktion „zoom fit“ bekommt ihr den besten Überblick.
 - o Ein Signal, das aus mehreren Bits besteht, kann man mit „Edit->Expand“ in einzelne Bits aufteilen.

Symbole für Gatter

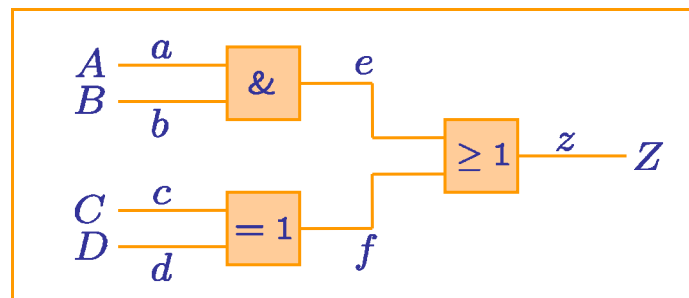
Symbol	Deutsche Bezeichnung	Englische Bezeichnung
	Inverter bzw. NICHT	NOT
	UND	AND
	Nicht-UND	NAND
	ODER	OR
	Nicht-ODER	NOR
	Exklusiv-ODER (Antivalenz)	XOR
	Exklusiv-Nicht-ODER (Äquivalenz)	XNOR

D-Algorithmus

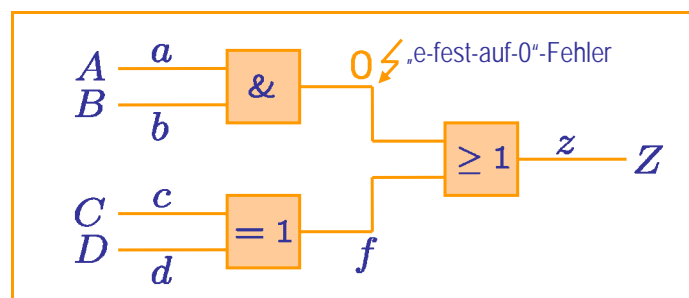
Das folgende einfache Verfahren, das als D-Algorithmus bezeichnet wird, kann zum Finden geeigneter Testmuster verwendet werden:

1. Verfolge den angenommenen Fehler zu den Ausgängen und finde heraus, welche Werte an den Eingängen anliegen müssen, sodass der Fehler am Ausgang sichtbar wird.
2. Stelle den Fehler durch geeignete Kombinationen an den Eingängen ein. Notiere die gefundenen Eingangskombinationen.
3. Prüfe die Ergebnisse aus 1. und 2. und wähle eine Eingangskombination aus, die in beiden Ergebnissen vorkommt.

Der D-Algorithmus wird anhand der folgenden einfachen Schaltung verdeutlicht:

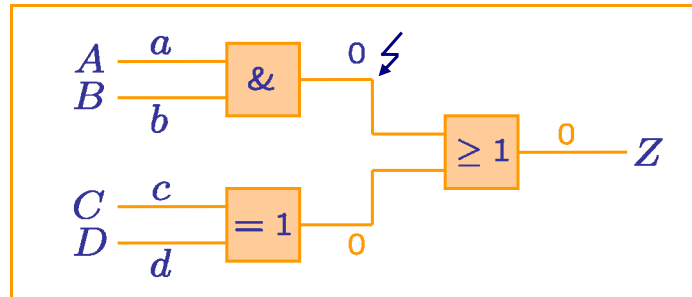


Im Folgenden nehmen wir den Fehler „e-fest-auf-0“ an, der gefunden werden soll.

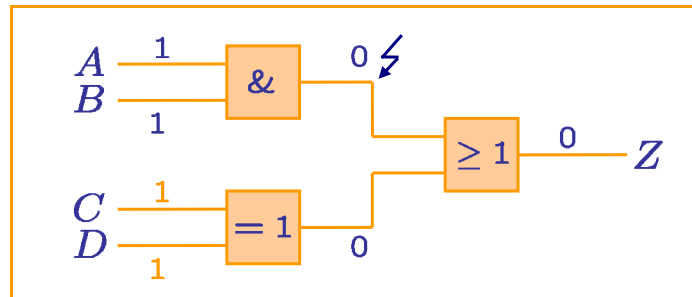


Schritt 1.1: Fehler zum Ausgang verfolgen. Hierzu müssen die Datenpfade (Signalwege) von dem fehlerhaften Signal zu den Ausgängen so geschaltet sein, dass der Unterschied zwischen fehlerfrei und fehlerhaft am Ausgang unterscheidbar ist. Damit ein Fehler an einem Eingang eines OR-Gatters unterscheidbar vom fehlerfreien Fall ist, muss am fehlerfreien Eingang des OR-Gatters eine 0 anliegen (Wenn ein Eingang bei einem OR-Gatter 1 ist, dann ist das Ergebnis unabhängig, ob ein Fehler auftritt oder nicht immer 1.):

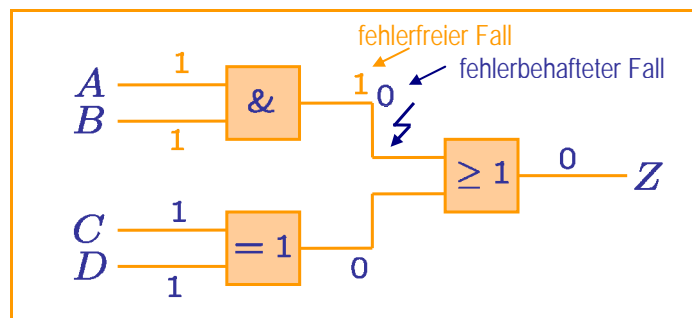
D-Algorithmus - Fortsetzung



Schritt 1.2: Nötiges Eingangsmuster finden, damit aus dem Exklusiv-ODER-Gatter eine 0 „herauskommt“: Die Eingänge A und B spielen für das Exklusiv-ODER-Gatter (kann 0 oder 1 sein) keine Rolle, da sie mit diesem Gatter nicht verbunden sind. Die Eingänge C und D müssen gleich sein (beide 0, oder beide 1). Aus diesem Grund ist die Wertebelugung 0000, 0100, 1000, 1100, 0011, 0111, 1011 und 1111, für die Eingänge A, B, C und D möglich. In der folgenden Abbildung erfolgte eine willkürliche Festlegung der Eingänge auf 1111:



Schritt 2: Fehler einstellen: Der zum angenommenen Fehler umgekehrte Wert für Signal e wäre e=1. Eine 1 als Ausgang von einem UND-Gatter erhalten wir, wenn beide Eingänge 1 sind. Hier spielen die Eingänge C und D keine Rolle. Die Eingänge A und B müssen also beide den Wert 1 haben. Das bedeutet, dass mögliche Eingangswerte 1100, 1101, 1110, und 1111 für ABCD sind, wobei in der folgenden Abbildung willkürlich die Belegung 1111 gewählt wurde:



Schritt 3: Eingangskombinationen finden, die bei den beiden Kombinationen aus Schritt 2 und 3 auftreten. Das sind 1100 und 1111 für ABCD. Beide Eingangskombinationen erzeugen bei einer fehlerhaften Schaltung eine 0 am Ausgang Z. Im fehlerfreien Fall wäre der Ausgang für diese Kombinationen 1!