

Praktikum "Grundlagen Technische Informatik"

Hochschule Osnabrück
Fakultät Ingenieurwissenschaften und Informatik
Labor für Digital- und Mikroprozessortechnik
Prof. Dr.-Ing. M. Weinhardt, Dipl.-Inf. (FH) R. Höckmann

WS20/21

Versuch 4: Praktischer Aufbau einer Schaltung mit realen Logikbausteinen und FPGAs

In diesem Versuch werden im letzten Versuch simulierte Schaltungen praktisch im Labor aufgebaut. Außerdem wird der Code-Umsetzer zur Ansteuerung einer Sieben-Segment-Anzeige durch einen Zähler ergänzt, welcher die anzuzeigenden Werte taktgesteuert erzeugt.

Für diesen Versuch muss keine Ausarbeitung erstellt werden!

Vorbereitung (Vor dem Praktikum)

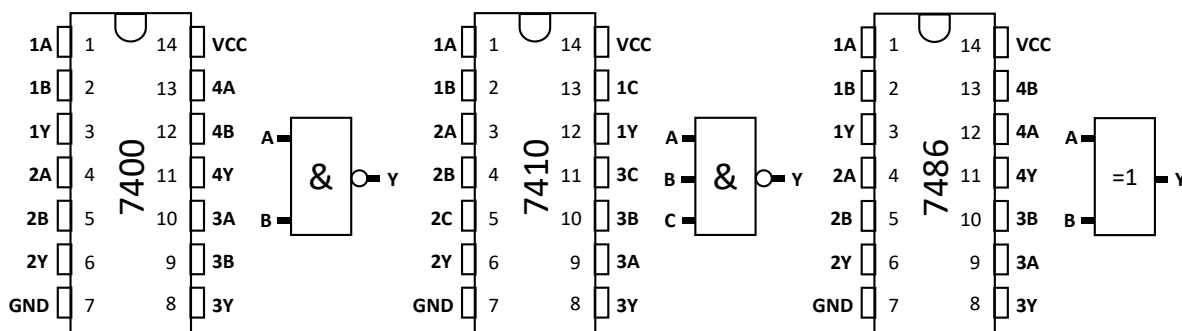
- Wiederholen Sie die Abschnitte der Vorlesung über die Realisierung digitaler Schaltungen.
- Wiederholen Sie die Abschnitte der Vorlesung über synchrone Zähler.
- Bringen Sie die Logisim-Evolution-Datei mit den erfolgreich getesteten Schaltungen zum Volladdierer und zum Code-Umsetzer aus Versuch 3 in das Labor mit. Sie können dazu zum Beispiel die hochschuleigene Cloud „Netcase“ oder einen USB-Stick verwenden.

Aufgabe 1: Praktischer Aufbau eines Volladdierers mit realen Logikbausteinen

Beim letzten Versuch haben Sie einen Volladdierer mit Logisim-Evolution erstellt und simuliert. Diese Schaltung soll heute praktisch mit realen Logikbausteinen aufgebaut werden

- Bauen Sie den Volladdierer aus den realen Logikbausteinen auf den Logiktrainer-Boards im Labor auf. Verbinden Sie dazu zunächst die Pins der Logik-Bausteine für die Versorgungsspannung (VCC) und für die Masse (GND) mit den entsprechenden Anschlüssen des Boards. Verbinden Sie die Eingänge der Schaltung mit den Schaltern des Boards und die Ausgänge mit LEDs.
- Lassen Sie vor dem Einschalten einen Versuchsbetreuer die Schaltung ansehen. Schalten Sie das Netzteil ein und testen Sie die Funktion der Schaltung. Beheben Sie eventuell vorhandene Fehler und führen Sie am Ende die korrekt funktionierende Schaltung vor.

Hier noch mal eine Übersicht über die Pin-Belegungen und die logischen Funktionen der verwendeten Logikbausteine 7400 (4x NAND2), 7410 (3x NAND3) und 7486 (4x XOR2):

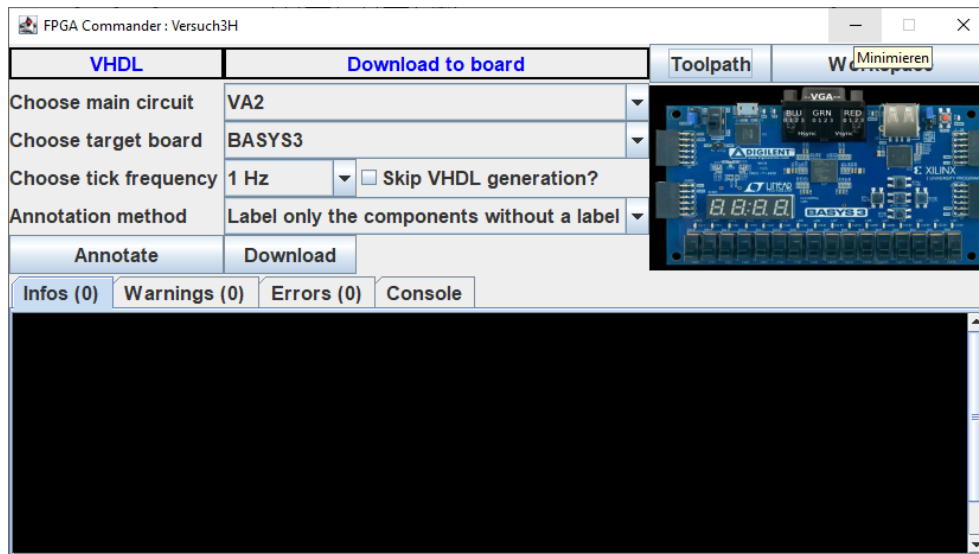


Aufgabe 2: Praktischer Aufbau eines Volladdierers mit einem FPGA

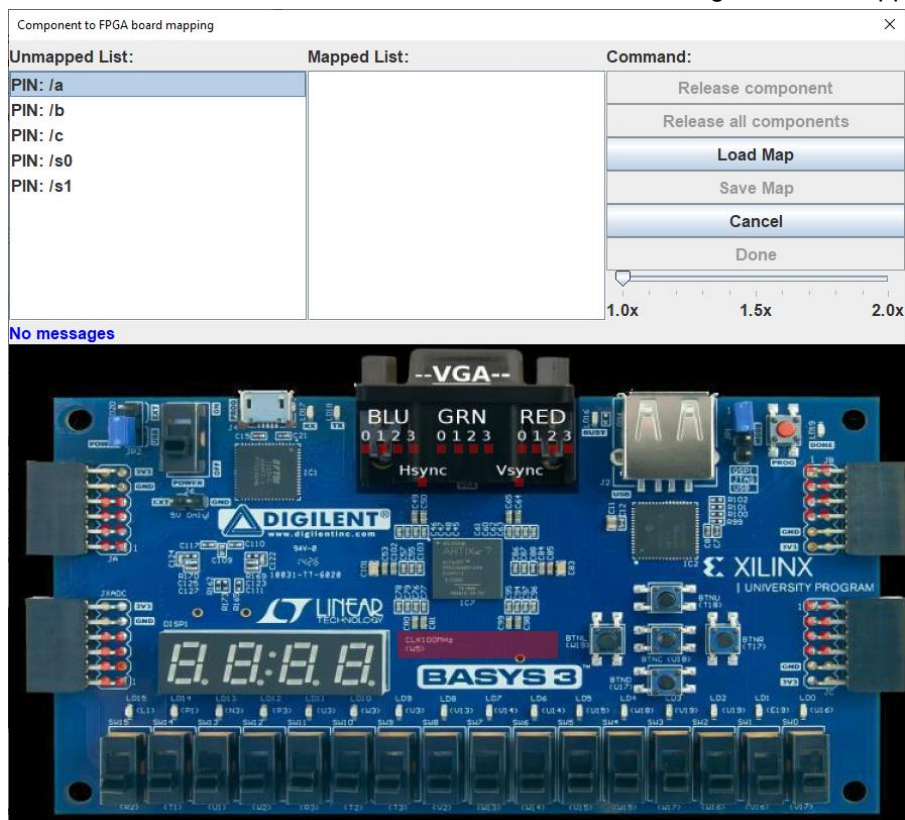
Digitalschaltungen werden heutzutage nur noch selten mit diskreten Logikbausteinen aufgebaut. In der Regel werden dafür anwendungsspezifische integrierte Schaltungen (ASIC) oder programmierbare Logikbausteine (FPGA) eingesetzt. Der Volladdierer wird nun mit einem FPGA realisiert.

- Starten Sie Logisim-Evolution und öffnen Sie damit die Datei **Versuch3.circ** aus dem letzten Versuch.

- b) Wählen Sie aus dem Menü „FPGAMenu“ die Option „FPGA Commander“, und stellen Sie als Hauptschaltung (main circuit) die Schaltung mit dem Volladdierer „VA2“ ein. Stellen Sie als Board (target board) „BASYS3“ ein und klicken Sie anschließend auf „Download“.

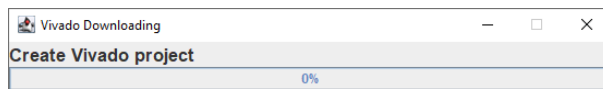


- c) In dem sich nun öffnenden Fenster „Component to FPGA board mapping“ müssen nun den Ein- und Ausgangsports der Schaltung passende Komponenten auf dem Board zugeordnet werden. Wählen Sie dazu nacheinander die Signale aus der „Unmapped List“ an und anschließend entweder Schiebeschalter für die Eingänge oder LEDs für die Ausgänge. Damit die Schalter und die Leuchtdioden des Boards auswählbar sind, müssen Sie die Signalnamen doppelt anklicken.

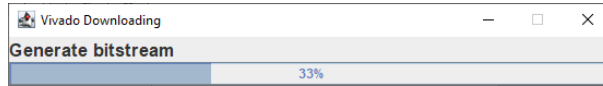


Die zugeordneten Signale erscheinen in der „Mapped List“. Sie können die Zuordnung mit „Save Map“ speichern. Klicken Sie anschließend auf „Done“.

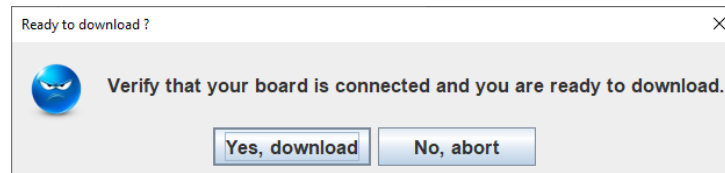
- d) Logisim-Evolution erzeugt nun eine Beschreibung des Systems in der Hardwarebeschreibungssprache VHDL und erzeugt ein dazugehöriges Projekt für das Synthese-Programm Vivado:



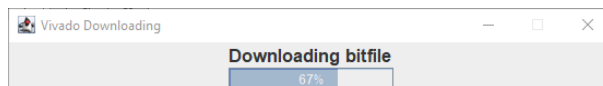
Anschließend wird Vivado (im Hintergrund) aufgerufen und aus der VHDL-Beschreibung eine Konfigurationsdatei für das FPGA erzeugt (Bitstream):



Nach erfolgreichem Abschluss dieses Vorgangs (der ein paar Minuten dauern kann) erscheint folgende Meldung:



Überprüfen Sie, ob das BASYS3-Board am Rechner angeschlossen und eingeschaltet ist (rote LED leuchtet). Klicken Sie anschließend den Button „Yes, download“:



Nach ein paar Sekunden leuchtet auf dem Board die grüne „DONE“-LED auf. Nun ist das FPGA programmiert.

- e) Überprüfen Sie nun für alle Kombinationen der Eingänge (Schiebeschaltern), ob die erwarteten Werte an den Ausgängen (LEDs) erscheinen. Demonstrieren Sie das funktionierende System den Praktikumsbetreuern.

Aufgabe 3: Realisierung des Code-Umsetzers auf einem FPGA

- Öffnen Sie die Logisim-Schaltung des Code-Umsetzers zur Ansteuerung einer Sieben-Segment-Anzeige. Sie soll nun ebenfalls mit einem FPGA realisiert werden. Rufen Sie den „FPGA-Commander“ auf und gehen wie in Aufgabe 1 vor. Klicken Sie dann auf „Download“.
- In der „Unmapped List“ sind wieder die noch nicht zugewiesenen Pins aufgeführt. Verbinden Sie die Eingabepins („BUS: /Eingabe#Pin...“) mit vier nebeneinanderliegenden Schiebeschaltern (mit Pin0 von rechts beginnend))



Die Anschlüsse der Siebensegmentanzeige werden mit den Anschlüssen der Siebensegmentanzeige verbunden:

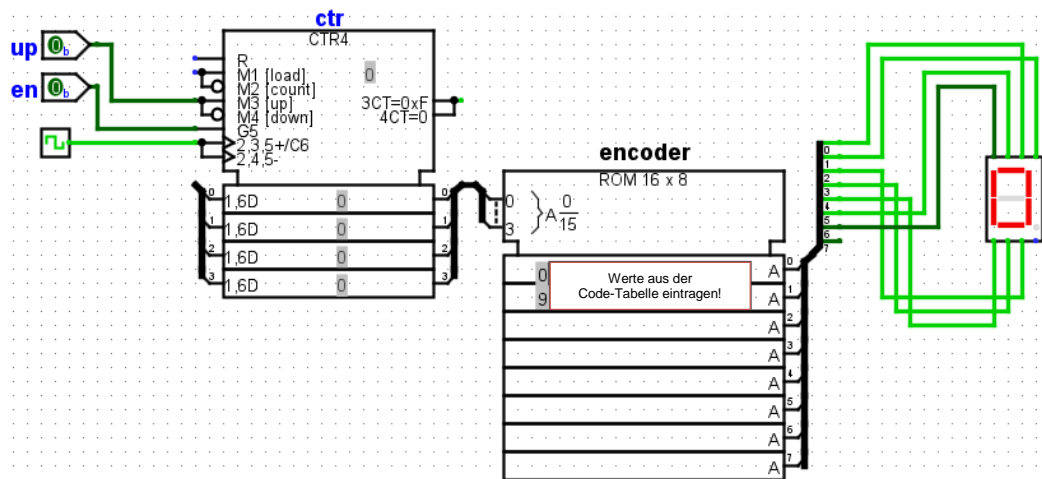


- Nun sollten Sie die Zuordnung abspeichern und mit „Done“ die Erzeugung des Bitstreams anstoßen. Danach programmieren Sie das FPGA-Board und demonstrieren das funktionierende System den Praktikumsbetreuern.

Aufgabe 4: Ergänzung um einen Synchronen Zähler

Das in Aufgabe 2 getestete System soll um einen Zähler ergänzt werden, der taktgesteuert Eingabewerte für die 7-Segmentanzeige erzeugt.

- Erzeugen Sie in Ihrem Logisim-Projekt eine neue Schaltung „Zaehler“. Kopieren Sie die Inhalte der Schaltung „Anzeige“ in die neue Schaltung „Zaehler“. Beim Kopieren werden die Beschriftungen entfernt. Benennen Sie das ROM wieder als „encoder“ und die 7-Segment-Anzeige als „sseg“.
- Entfernen Sie den Eingangs-Pin. Instanzieren Sie einen „Zähler“ (CTR4, Rubrik: Speicher) und setzen Sie dessen Attribut „Datenbits“ auf den Wert 4. Verbinden Sie den Datenausgang des Zählers mit dem Adresseingang des ROMs.
- Als synchrone Komponente benötigt der Zähler einen Takteingang. Deshalb instanzieren Sie auch noch einen „Takt“ (Puls, Rubrik: Verdrahtung) und verbinden ihn mit dem Takteingang C6 des Zählers.
- Zur Steuerung des Zählers dienen die Eingänge G5 und M3. Eine 1 an G5 sorgt dafür, dass der Zähler seinen Wert bei jeder aktiven Taktflanke verändert (Freigabe). Mit dem Eingang M3 wird die Zählrichtung eingestellt (1=Auf, 0=Ab). Zur Eingabe dieser beiden Werte verwenden Sie Eingangs-Pins. Benennen Sie die Eingänge als „up“ und „en“. Das System könnte jetzt etwa so aussehen:



- Starten Sie nun den simulierten Takt (Simulieren→Weiterschalten aktivieren). Die Schaltfrequenz des Simulationstakts kann geändert werden (Simulieren→Schaltfrequenz). Mit dem Hand-Werkzeug können Sie die Schalter betätigen. Testen Sie, ob sich das System wie erwartet verhält.
- Übertragen Sie das System auf das FPGA wie bereits in Aufgabe 1 und 2 beschrieben. Wählen Sie im Fenster „FPGA Commander“ als „tick frequency“ den Wert „2 Hz“ aus. Logisim fügt in das System später automatisch einen Taktteiler ein, um aus dem 50 MHz-Takt des BASYS3-Boards das 2 Hz-Taktsignal zu erzeugen. Verwenden Sie zwei Schiebeschalter für die Steuersignale. Die Verbindung der 7-Segmentanzeige erfolgt wie bereits in Aufgabe 1 beschrieben.
- Demonstrieren Sie das funktionierende System den Praktikumsbetreuern.