

Wecker

Entwerfen Sie einen Wecker mit 24h-Uhrzeit-Anzeige und einer Alarmzeit sowie Snooze-Funktion. Der Alarm soll sie nicht aus dem Schlaf reißen, sondern durch langsam aufleuchtende LEDs sanft aufwecken.

a) Uhrwerk mit Funkuhr

Im Normalzustand soll die 7-Segment Anzeige des Boards die Uhrzeit im 24h HH:MM Format anzeigen und ein Dezimalpunkt im Sekundentakt blinken. Ein Schalter soll die Anzeige umschalten und soll Minuten und Sekunden im Format MM:SS ausgeben.

Der Funkuhrempfänger aus den Übungsaufgaben soll als Zeitquelle für die Uhrzeit genutzt werden. Die Uhrzeit soll dennoch auch manuell einstellbar sein.

Mit einem Schalter wird die Eingabe der Uhrzeit und mit einem anderen Schalter die Eingabe der Alarmzeit aktiviert. Die Anzeige beginnt dabei zu blinken und die Uhr- bzw. Alarmzeit wird angezeigt. Beim Einstellen der Uhrzeit bleibt das Uhrwerk stehen. Es soll immer nur genau eine Zeit einstellbar sein.

Die Zeit soll mit den vier Buttons eingestellt werden können:

Je zwei Buttons werden zum Schrittweisen hoch- und herunterzählen jeweils für Stunde und Minute genutzt (also die vier Buttons mit der Funktion: Stunde+, Stunde-, Minute+, Minute-).

Die Änderung der Minute soll keinen Einfluss auf die Stunden haben und umgekehrt. Während des Einstellens darf kein Alarm ausgelöst werden.

Wird eine Taste gedrückt gehalten, dann soll nach einem kurzen Moment die Tastenwiederholung aktiviert werden, um die Zeit schneller einstellen zu können. Wählen Sie eine angemessene Wartezeit und Tasten-Wiederholungsfrequenz.

Realisieren Sie ein "Uhrwerk"-Modul derart, dass das gleiche Modul in zwei Instanzen, sowohl für die laufende Uhr als auch zum Einstellen der Alarmzeit genutzt werden kann. Das Modul speichert die Zeit und enthält das hoch/runter Zählwerk mit der Einstellen-Funktion. Es soll mit dem 50MHz Takt betrieben werden und über einen Reset verfügen, der die Zeit anhält und auf null setzt. Allerdings enthält es keinen Taktgeber für das Zählwerk oder Zeitmessung für die Tastenwiederholung.

Soll das Uhrwerk laufen, dann muss das Weiterzählen der Sekunden durch einen kurzen Enable-Strobe aktiviert wird. Die Einstellung der Zeit erfolgt ebenfalls über solche Strobe-Signale.

Die gespeicherte Zeit soll vom Modul als output für andere Schaltungsteile zur Verfügung gestellt werden.

b) Wecker

Die Weckfunktion soll über einen eigenen Schalter aktiviert werden. Ist die eingestellte Alarmzeit erreicht, dann wird der Alarm aktiv und weckt durch das Einschalten eines Lichts (alle sieben LEDs).

Der Alarm soll automatisch nach 30 Sekunden Aktivität abgeschaltet werden oder sofort, wenn der Wecker-Schalter deaktiviert wird.

Realisieren Sie die Ablaufsteuerung des Alarms in Form einer FSM in eigenem Verilog-Modul. Nutzen Sie das "casex ({inputvector, current_state})" Format zur Abbildung der Zustandübergänge. Das Modul soll ausschließlich die FSM und deren Übergänge enthalten. Alle anderen Funktionen müssen in andere (Unter-)Module ausgelagert werden und über Steuersignale mit der FSM interagieren.

c) Optional: Sanftes Wecken und Snooze

Durch das schlagartige Einschalten des Lichts wird man unsanft aus dem Schlaf gerissen. Implementieren Sie die Funktion "sanftes Wecken":

Mit ihr sollen die LEDs nicht schlagartig eingeschaltet werden, sondern eine nach der anderen langsam heller werden. Hat eine LED die volle Helligkeit erreicht, dann beginnt die nächste langsam aufzuleuchten. Beim Ausschalten sollen die LEDs gemeinsam dunkel werden.

Nutzen Sie Pulsweitenmodulation (PWM), um die Helligkeit zu steuern. Jede LED soll eine eigene Instanz des PWM-Moduls zugewiesen bekommen.

Erweitern Sie die Wecker-FSM um die Snooze-Funktion:

Während des Alarms kann durch das Drücken eines Buttons die Snooze-Funktion genutzt werden, um nochmals 10 Sekunden Ruhe zu haben. Danach wird der Alarm wieder 30 Sekunden lang aktiv. Snooze soll maximal 3x genutzt werden dürfen.

d) Optional: PS/2 Tastatur

Entwerfen Sie ein Modul, das mit der Tastatur kommunizieren kann. Es soll sowohl Tastencodes empfangen können, als auch die LEDs der Tastatur ansteuern.

Das Einstellen von Uhr- und Weckzeit soll über die Zifferntasten der PS/2-Tastatur zusätzlich zu dem Einstellen per Buttons ermöglicht werden. Die 4 Ziffern werden nacheinander eingelesen und gleich auf der 7-Segment-Anzeige angezeigt. Der Eingabemodus für die Uhrzeit und die Alarmzeit wird über F-Tasten aktiviert und deaktiviert. Die Snooze-Funktion soll ebenfalls per Tastendruck ausgelöst werden können.

Die aktivierte Weckfunktion soll über eine LED auf der Tastatur angezeigt werden. Ein aktiver Alarm soll die drei LEDs der Tastatur nacheinander einschalten und dann blinken lassen.

Verwenden Sie Zustandsmaschine(n), um die Zustände der PS/2-Protokolls sowie der Eingabe und LED Ansteuerung abzubilden.

Verilog-Code und Abschlussbericht

- Der Verilog-Code und der Abschlussbericht sind von Ihnen alleine anzufertigen. Gruppenarbeit ist nicht zulässig.
- Der Verilog-Code muss für das Nexys2 Board synthetisierbar sein. Im Verilog-Code müssen Kommentare enthalten sein, die die Funktion grob erläutern. Benennen Sie mit "sprechenden" Bezeichnern, um den Code verständlicher zu machen.
- Alle Ihre Module müssen mit der 50MHz Taktfrequenz oder durch mit Takteilern davon abgeleiteten Taktsignalen betrieben werden und ein Reset verwenden.
- Alle Module müssen von Ihnen selbst geschrieben sein. Die Verwendung von Teilen aus den Musterlösungen der Übung ist nicht zulässig.
- Der Bericht soll die Funktion Ihres Verilog-Codes erläutern. Es soll ersichtlich werden, dass Sie den Code selbst geschrieben und verstanden haben. Sie sollen die Arbeitsweise der Module und deren Interaktion beschreiben. Erläutern Sie in ihrem Bericht exemplarisch an einer Ihrer Zustandsmaschinen, warum Sie diese so aufbauen und motivieren Sie die Art, wie Sie sie codiert haben.
- Geben Sie alle verwendeten Quellen an.

- Der Bericht sollte nicht länger als 10 Seiten sein.
 - Bei belegten Täuschungsversuchen wird sowohl der kopierende als auch der kopierte Prüfling mit einer 5.0 bewertet werden.
 - Geben Sie alle Verilog-Dateien (.v), die UCF-Datei, den Bericht (in PDF-Format) und den zweifach unterzeichnete Versicherungs-Passus (PDF-Scan; Text siehe unten) ab.
 - Abgabe spätestens am 17.04.2023 per Email an christian.kreidl@ziti.uni-heidelberg.de
- Versicherung

Ich versichere die vorliegende Prüfungsleistung selbständig verfasst und keine anderen als die erlaubten Hilfsmittel benutzt zu haben. Ich habe während der Bearbeitung der Aufgaben in keinerlei Weise mit anderen Personen oder elektronischen Medien über die Aufgaben, mögliche Lösungen und ähnliches kommuniziert. Bei Abgabe einer unwahren Versicherung wird die Prüfung mit "nicht bestanden" bewertet.

Ort, Datum Unterschrift

Bei Verdacht auf Abgabe einer unwahren Versicherung können die Prüflinge vom Prüfer zu einer mündlichen Kontrolle einbestellt werden.

Ort, Datum Unterschrift

Bewertungskriterien

- Ein lauffähiger Code, der die minimalen Anforderungen umsetzt, zusammen mit einem aussagekräftigen Bericht, qualifiziert für eine Note 2.3. Für eine 1.0 sind zusätzlich Teilaufgabe c) und die Tastatureingabe nötig.
- Werden Vorgaben nicht erreicht oder umgesetzt, ist der Code unstrukturiert oder ungeschickt oder fehlerhaft geschrieben, verschlechtert sich die Note.
- Umgekehrt wird die Note verbessert durch gut strukturierten Code, durch eine übersichtliche und gut kommentierte Implementierung, durch einen sehr klaren Bericht, durch gut gewählte Zustandsmaschinen und durch die Implementierung der zwei optionalen Zusatzaufgaben.