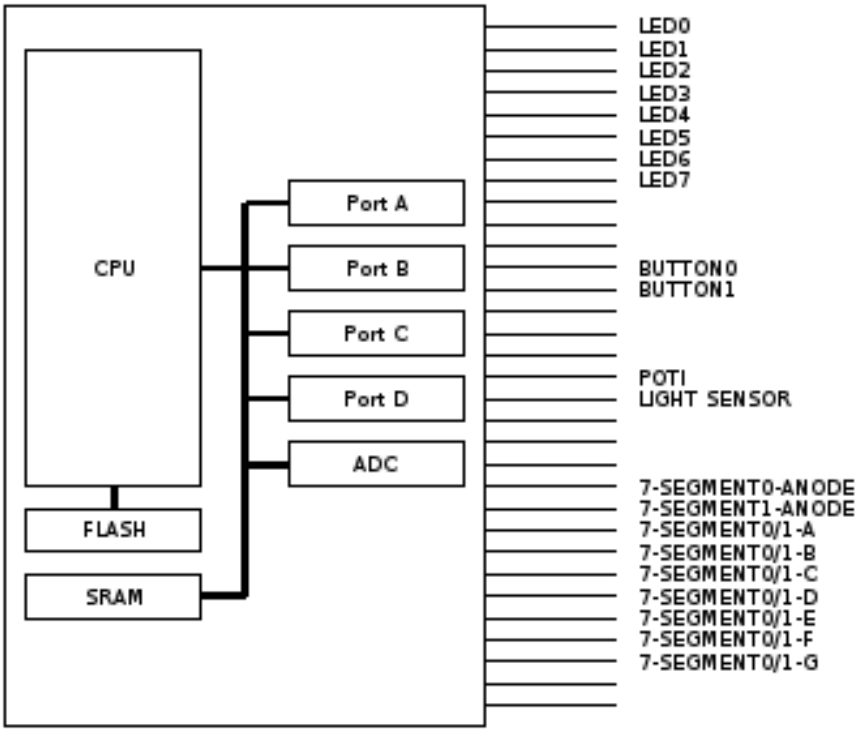
Übungen zu Virtuelle Maschinen, Aufgabe 2

Volkmar Sieh

WS2018/2019

1 Übersicht

In Aufgabe 2 soll die in Aufgabe 1 entwickelte virtuelle Maschine dahingehend erweitert werden, dass die I/O-Ports sowie der AD-Wandler funktionieren.



Die Aufgabe gilt als erfüllt, wenn die Test-Programme „adc“, „button“, „led“ und „seg7“ korrekt ausgeführt werden.

2 Organisatorisches

Bitte senden Sie Ihre Lösung bis zum 15.12.2018 per Mail an i4vm@cs.fau.de. Die Testprogramme finden Sie unter
https://www4.cs.fau.de/Lehre/WS18/V_VM/Uebungen/.

3 Hardware-Simulatoren

3.1 I/O-Ports

Im ATmega32 sind 4 8-Bit-General-Purpose-I/O-Ports integriert. Diese können sowohl für die Ausgabe wie auch für die Eingabe genutzt werden. Ihre Implementierung soll beide Modi unterstützen. Zu implementieren sind daher die Register DDRx (Data Direction Register), PORTx (Daten-Ausgabe-Register bzw. Pull-Up-Konfigurationsregister) sowie PINx (Data In Register).

Die sogenannten „Alternate Functions“ der Ports müssen nicht implementiert werden (Ausnahme: AD-Wandler; s.u.).

3.2 AD-Wandler

Der AD-Wandler im ATmega32 kann über ein Register ADMUX auswählen, an welchem der 8 Pins des Ports A des Mikrokontrollers die Spannung gemessen werden soll. Über ein Kontroll-Register (ADCSRA) lässt sich der ADC konfigurieren und starten. Das gleiche Register ist auch Statusregister, über das sich ermitteln lässt, ob eine AD-Wandlung abgeschlossen wurde. Das eigentliche Messergebnis kann aus dem Register ADCW ausgelesen werden.

Diese Register sind zu implementieren, soweit sie vom Testprogramm „adc“ gebraucht werden.

4 Kabelverbindungen

4.1 sig_std_logic

Das `sig_std_logic`-Modul enthält eine Abstraktion eine „Kabels“. Über diese Kabel können Spannungswerte übertragen werden. Ausgaben des Mikrokontrollers können z.B. an die Komponenten (LED, 7-Segment-Anzeige) weitergeleitet werden. Spannungen, die das Potentiometer, der Lichtsensor oder die Taster einstellen, gelangen darüber in den Mikrokontroller.

Über die Funktion `sig_std_logic_connect_in` kann man Callback-Funktionen an den Signalen registrieren, die aufgerufen werden, sobald sich die Spannung am Signal ändern. In der Callback-Funktion kann aus dem übergebenen Signalwert `val` mit `SIG_mV(val)` die Spannung (in mV) ermittelt werden.

Möchte eine Komponente auf einem Signal eine Spannung ausgeben, so muss die Komponente sich vorher mit `sig_std_logic_connect_out` am Signal registrieren. Später kann sie dann mit `sig_std_logic_set` Werte ausgeben. Vordefinierte Werte sind

`SIG_STD_LOGIC_0`: logische „0„ (0 Volt)

`SIG_STD_LOGIC_1`: logische „1„ (5 Volt)

`SIG_STD_LOGIC_Z`: es wird nichts ausgegeben