

Systemarchitektur SS 2021

Aufgabenblatt 8

Sie können Ihre Lösungen bis **Mittwoch, dem 16.06.2021, um 10:00 Uhr** im CMS abgeben.
Geben Sie auf Ihrer Lösung Ihr Tutorium sowie die Namen und Matrikelnummern aller Gruppenmitglieder an.

Aufgabe 8.1: Register-Swap

Schreiben Sie ein MIPS-Assembler-Programm, das die in Register R2 und R3 gespeicherten Werte vertauscht. Verwenden Sie hierfür keine anderen als die beiden genannten Register. Verwenden Sie auch keine Instruktionen, die den Speicher lesen oder beschreiben.

Aufgabe 8.2: Fibonacci-Zahlen

Schreiben Sie ein MIPS-Assembler-Programm, das die n -te Fibonacci-Zahl berechnet. Sie können annehmen, dass der Parameter n zu Beginn des Programms in Register `$a0` steht. Das Ergebnis des Programms soll in Register `$v0` gespeichert werden.

Aufgabe 8.3: MIPS: Datenpfad

Im Folgenden betrachten wir Einzeltakt-MIPS-Prozessoren (gemäß der Implementierung auf Foliensatz 12), die bestimmte Defekte haben. Geben Sie jeweils an, welche Instruktionen dadurch nicht mehr korrekt ausgeführt werden.

1. Nehmen Sie an, dass das *RegDst*-Signal einen sogenannten “*stuck-at-0*”-Fehler hat, d.h. das Signal ist immer 0.
2. Nehmen Sie an, dass das *MemWrite*-Signal einen “*stuck-at-0*”-Fehler hat.
3. Nehmen Sie an, dass das *RegWrite*-Signal einen “*stuck-at-1*”-Fehler hat.

System Architecture SS 2021

Assignment 8

You may submit your solutions via the CMS until **10:00 a.m. on Wednesday, June 16, 2021**.

Please state on your solutions your tutorial, and the names and matriculation numbers of all team members.

Problem 8.1: Register Swap

Write a MIPS assembler program that swaps the values stored in the registers R2 and R3. Do not use any other registers. Also, do not use instructions that read from or write to memory.

Problem 8.2: Fibonacci Numbers

Write a MIPS assembler program that computes the n -th Fibonacci number. You can assume that the parameter n is in register \$a0 at the start of the program. The result of the program shall be stored in register \$v0.

Problem 8.3: MIPS: Datapath

In the following, we consider single-cycle MIPS processors (according to the implementation on slide deck 12) that have specific faults. For each case, state which instructions are no longer executed correctly.

1. Assume that the *RegDst* signal has a so-called “*stuck-at-0*” fault, i.e., the signal is always 0.
2. Assume that the *MemWrite* signal has a “*stuck-at-0*” fault.
3. Assume that the *RegWrite* signal has a “*stuck-at-1*” fault.