

3. Übung für die Vorlesung Rechnerorganisation

Sommersemester 2019

Abgabe: Donnerstag, 25.04.2018; Schicken Sie bitte den Quellcode für die Programmieraufgabe zusätzlich per E-Mail an Ihren Tutor:

Alexandra Chebotareva: `s6alcheb@uni-bonn.de`
 Michel Fischer: `michel-fischer@hotmail.de`

Aufgabe 1. *MIPS-XOR*

4 P.

Die MIPS ISA beinhaltet den Befehl `xor` ($a \oplus b := ab' + a'b$). Überlegen Sie, wie Sie diesen Befehl nutzen können um zwei Registerinhalte zu vertauschen, ohne ein drittes Register zu verwenden. Geben Sie die Befehlssequenz an, welche den Tausch der Registerinhalte von `$s1` und `$s2` auf diese Weise realisiert. Beweisen Sie, dass Ihre Befehlssequenz funktioniert.

Aufgabe 2. *MIPS-ISA*

12 P.

Gegeben seien die folgenden MIPS-Instruktionen:

`and, or, xor, nor, andi, ori, xori, nori, add, addi, slt, sll, srl, beq, bne, j`

Realisieren Sie ausschließlich mit deren Hilfe die folgenden "Pseudo" -Befehle:

1	<code>sbi</code>	<code>\$d,</code>	<code>\$s,</code>	<code>imm</code>	Setze das <code>imm</code> -te Bit vom Inhalt von <code>\$s</code> auf '1' und speichere das Ergebnis nach <code>\$d</code> (Least-Significant-Bit ist Bit 0)
2	<code>rbi</code>	<code>\$d,</code>	<code>\$s,</code>	<code>imm</code>	Setze das <code>imm</code> -te Bit vom Inhalt von <code>\$s</code> auf '0' und speichere das Ergebnis nach <code>\$d</code>
3	<code>tbi</code>	<code>\$d,</code>	<code>\$s,</code>	<code>imm</code>	Invertiere das <code>imm</code> -te Bit vom Inhalt von <code>\$s</code> und speichere das Ergebnis nach <code>\$d</code>
4	<code>sra</code>	<code>\$d,</code>	<code>\$s,</code>	<code>shamt</code>	schiebe den Inhalt von <code>\$s</code> um <code>shamt</code> -Stellen <u>arithmetisch</u> nach rechts und speichere das Ergebnis nach <code>\$d</code>
5	<code>sllv</code>	<code>\$d,</code>	<code>\$s,</code>	<code>\$t</code>	schiebe den Inhalt von <code>\$s</code> um <code>\$t</code> -Stellen nach links und speichere das Ergebnis nach <code>\$d</code>

Für Zwischenergebnisse steht Ihnen das Register `$at` zur Verfügung, andere Registerinhalte (ausgenommen `$d`) dürfen nicht verändert werden.

Aufgabe 3. *MIPS-Assembler Fakultät*

6 P.

Schreiben Sie ein Programm für die Fakultätsberechnung einer natürlichen Zahl aus $[0..12]$. In Ihrem Programm soll eine Überprüfung der Eingabe auf Gültigkeit stattfinden. Wird ein ungültiger Wert eingegeben, so soll als Fakultätswert -1 ausgegeben werden. Die Eingabe N steht im Register `$s0`, das Ergebnis $N!$ in `$s1`.

Benutzen Sie zum Testen Ihres Programms, das auf unserer Webseite vorgegebene MIPS Programm `fakultaet.asm`. Fügen Sie dazu Ihr Programm in der dafür vorgesehenen Stelle ein und führen es mit SPIM aus.

Hinweis: Sie können für die Multiplikation zweier Registerinhalte die folgende Instruktion verwenden:

`mul $rd, $rs, $rt` # Put the low-order 32 bits of the product of `$rs` and `$rt` into register `$rd`.