
Übungsblatt 4

Ü 4.1 RTL-Beschreibung

Geben Sie die RTL-Beschreibung (register transfer language) für folgende MIPS-Instruktionen an: `add`, `ori`, `lw`, `bne`, `jal`.

Ü 4.2 Single-Cycle Datenpfad: `sw`-Instruktion erklären

Erklären Sie die `sw`-Instruktion für den Single-Cycle-Datenpfad, der in der Vorlesung vorgestellt wurde. Geben Sie dazu zuerst die RTL-Beschreibung der Instruktion an. Geben Sie die Werte aller Kontrollsignale an, und kennzeichnen Sie Don't Care Werte mit „X“.

Ü 4.3 Single-Cycle Datenpfad: `addsw`-Instruktion

Erweitern Sie den Single-Cycle Datenpfad aus der Vorlesung um die (fiktive) „add store word“-Instruktion: `addsw $rd, $rs, $rt` (R-Format). Diese Instruktion berechnet die Summe der Registerinhalte von `$rs` und `$rt` und schreibt die Summe an die Speicheradresse, die im Register `$rd` enthalten ist.

- (a) Geben sie die RTL-Beschreibung für diese Instruktion an.
- (b) Führen Sie alle notwendigen Änderungen am Single-Cycle-Datenpfad durch, um diese Instruktion zu realisieren. Geben Sie alle Steuersignale, insbesondere auch möglicherweise zusätzlich benötigte, an. Verwenden Sie dafür die zur Verfügung gestellten Abbildungen. Don't care Werte müssen mit „X“ bezeichnet werden.
- (c) Begründen Sie, warum diese Instruktion nicht den Opcode 0 haben darf (wie andere R-Format Befehle).

Ü 4.4 Single-Cycle Datenpfad: `jrne`-Instruktion

Erweitern Sie den Single-Cycle-Datenpfad aus der Vorlesung um die (fiktive) „jump to register on not equal“-Instruktion (`jrne $rd, $rs, $rt`). Diese Instruktion springt an die Adresse in `$rd`, wenn der Inhalt von `$rs` ungleich dem Inhalt von `$rt` ist. Das Befehlsformat ist mit dem eines R-Format-Befehls identisch.

- (a) Geben Sie die RTL-Beschreibung für diese Instruktion an.
 - (b) Führen Sie alle notwendigen Änderungen am Single-Cycle-Datenpfad durch, um diese Instruktion zu realisieren. Geben Sie alle Steuersignale, insbesondere auch möglicherweise zusätzlich benötigte, an. Verwenden Sie dafür die zur Verfügung gestellten Abbildungen. Don't care Werte müssen mit „X“ bezeichnet werden.
-

Opcode	ALUOp	Operation	Funct field	ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	and	0000
R-type	10	OR	100101	or	0001
R-type	10	set on less than	101010	set on less than	0111

In: D. Patterson/J. Hennessy, Computer Organization and Design, 3rd ed., Fig. 5.12

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1