



Extrablatt Informatik II
(korrigierte Version 1.8)

Alle imm-Konstanten sind, wie in der Vorlesung gesagt, sign extended.

I-type Instructions

I[31:26]	mnemonic	effect
load/store		
100011	lw	$RD = m(RS1 + sxt(imm))$
101011	sw	$m(RS1 + sxt(imm)) = RD$
01****	comp.imm	$RD = aluop(I[29:26])(RS1, imm)$
control		
110100	beqz	$PC = PC + (RS1 = 0 ? imm:1)$
110101	bnez	$PC = PC + (RS1 \neq 0 ? imm:1)$
110110	jr	$PC = RS1$
110111	jalr	$R31=PC+1 ; PC = RS1$

R-type Instructions

I[31:26]	I[5:0]	mnemonic	effect
000000	00****	compute	$RD = aluop(I[3:0])(RS1, RS2)$
special moves			
000000	010000	movs2i	$SA = RS1 \quad (SPR[SA] \rightarrow GPR[RS1])$
000000	010001	movi2s	$RD = SA \quad (GPR[RD] \rightarrow SPR[SA])$

J-type Instructions

I[31:26]	mnemonic	effect
control		
000010	j	$PC = PC + imm$
000011	jal	$R31=PC+1 ; PC = PC + imm$
111110	trap	spaeter
111111	rfe	spaeter



Extrablatt Informatik II
 (korrigierte Version 1.8)

I-type comp.imm

I[31:26]	mnemonic	effect
01 ****	comp.imm	$RD = \text{aluop}(I[29:26])(RS1, \text{imm})$
Arithmetic, Logical, Operation		
01 0000	addi	$RD = RS1 + \text{imm}$
01 0001	addi	$RD = RS1 + \text{imm}$ (no overflow)
01 0010	subi	$RD = RS1 - \text{imm}$
01 0011	subi	$RD = RS1 - \text{imm}$ (no overflow)
01 0100	andi	$RD = RS1 \wedge \text{sxt}(\text{imm})$
01 0101	ori	$RD = RS1 \vee \text{sxt}(\text{imm})$
01 0110	xori	$RD = RS1 \oplus \text{sxt}(\text{imm})$
01 0111	lhgi	$RD = \text{imm} \cdot 0^{16}$
Test Set Operation		
01 1000	clri	$RD = (\text{false} ? 1 : 0);$
01 1001	sgri	$RD = (RS1 > \text{imm} ? 1 : 0);$
01 1010	seqi	$RD = (RS1 = \text{imm} ? 1 : 0);$
01 1011	sgei	$RD = (RS1 \geq \text{imm} ? 1 : 0);$
01 1100	slsi	$RD = (RS1 < \text{imm} ? 1 : 0);$
01 1101	snei	$RD = (RS1 \neq \text{imm} ? 1 : 0);$
01 1110	slei	$RD = (RS1 \leq \text{imm} ? 1 : 0);$
01 1111	seti	$RD = (\text{true} ? 1 : 0);$



Extrablatt Informatik II
(korrigierte Version 1.8)

R-type compute

I[31:26]	I[5:0]	mnemonic	effect
000000	00 ****	compute	RD = aluop(I[3:0])(RS1, RS2)
Arithmetic, Logical, Operation			
000000	00 0000	add	RD = RS1 + RS2
000000	00 0001	add	RD = RS1 + RS2 (no overflow)
000000	00 0010	sub	RD = RS1 - RS2
000000	00 0011	sub	RD = RS1 - RS2 (no overflow)
000000	00 0100	and	RD = RS1 \wedge RS2
000000	00 0101	or	RD = RS1 \vee RS2
000000	00 0110	xor	RD = RS1 \oplus RS2
000000	00 0111	lhg	RD = RS2[15:0] 0 ¹⁶
Test Set Operation			
000000	00 1000	clr	RD = (false ? 1 : 0);
000000	00 1001	sgr	RD = (RS1 > RS2 ? 1 : 0);
000000	00 1010	seq	RD = (RS1 = RS2 ? 1 : 0);
000000	00 1011	sge	RD = (RS1 \geq RS2 ? 1 : 0);
000000	00 1100	sls	RD = (RS1 < RS2 ? 1 : 0);
000000	00 1101	sne	RD = (RS1 \neq RS2 ? 1 : 0);
000000	00 1110	sle	RD = (RS1 \leq RS2 ? 1 : 0);
000000	00 1111	set	RD = (true ? 1 : 0);