Mr. Chips Greencard

Megan Notkin
Anna Weisman

May 2023

- Add: $R_{rd} = R_{rs} + R_{rt}$
- Subtract: $R_{rd} = R_{rs} - R_{rt}$
- Mul: R_{mfhi} = more significant bits of $R_{rs} * R_{rt}$; R_{mflo} = least significant bits of $R_{rs} * R_{rt}$
- Div: $R_{mfhi} = R_{rs}\%R_{rt}$; $R_{mflo} = R_{rs}/R_{rt}$
- And: $R_{rd} = R_{rs}\&R_{rt}$
- SLT: $R_{rd} = 1$ if $R_{rs} < R_{rt}$ else 0
- Jr: $PC = R_{rs}$
- Lw: $R_{rt} = M[R_{rs}$+SignExtImm]
- Sw: $M[R_{rs} + SignExtImm] = R_{rt}$
- Beq: if$(R_{rs} == R_{rt})$ $PC = PC + 1 + BranchAddr$
- Addi: $R_{rt} = R_{rs} + SignExtImm$
- J: $PC = JumpAddr$
- Jal: $R_7 = PC + 2; PC = JumpAddr$
- SLTI: $R_{rt} = 1$ if $R_{rs} < imm$ else 0

Where SignExtImm is defined as $9immediate[6], imm$, JumpAddr is defined as $(PC + 1)[15 : 13], address$, and BranchAddr is defined as $7immediate[6], immediate, 1'b0$.

	MSb															LSb
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R-TYPE	opcode			Rs			Rt			Rd			funct			
I -YPE	opcode			Rs			Rt			7-bit immediate						
J-Type	opcode			13-bit address												

	MSb															LSb
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	opcode			Rs			Rt			Rd			funct			
add	0	0	0	x	x	x	x	x	x	x	x	x	0	0	0	0
sub	0	0	0	x	x	x	x	x	x	x	x	x	0	0	0	1
mult	0	0	0	x	x	x	x	x	x	x	x	x	0	0	1	0
nor	0	0	0	x	x	x	x	x	x	x	x	x	0	0	1	1
slt	0	0	0	x	x	x	x	x	x	x	x	x	0	1	0	0
div	0	0	0	x	x	x	x	x	x	x	x	x	0	1	0	1
mfhi	0	0	0	x	x	x	x	x	x	x	x	x	0	1	1	0
mflo	0	0	0	x	x	x	x	x	x	x	x	x	0	1	1	1
jr	0	0	0	x	x	x	x	x	x	x	x	x	1	0	0	0

	MSb															LSb
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I-Type	opcode			Rs			Rt			7-bit immediate						
LOAD	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x
STORE	1	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x
ADDI	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x
SLTI	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x
BEQ	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x

	MSb															LSb
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
J-Type	opcode			(13-bit) address												
Jump	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x
JAL	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x