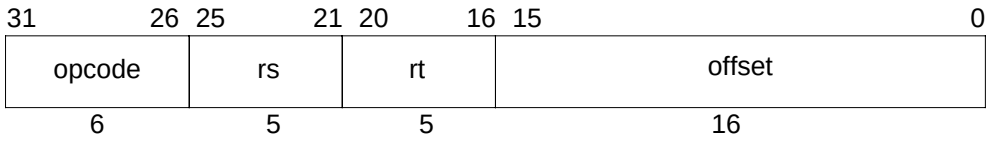
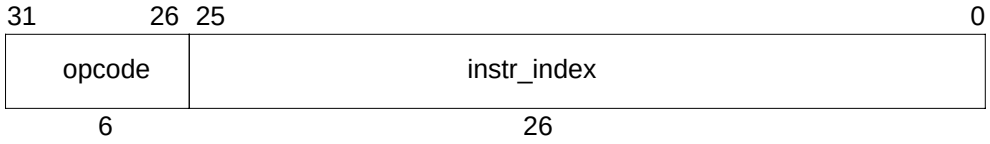
A 7 CPU Instruction Formats

A CPU instruction is a single 32-bit aligned word. The major instruction formats are shown in Figure A-10.

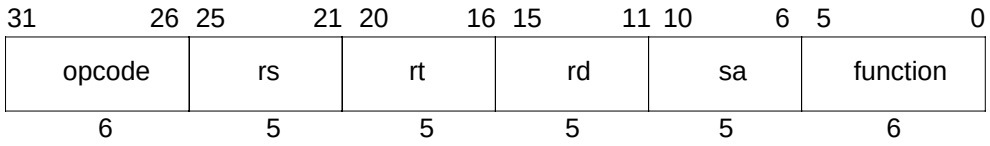
I-Type (Immediate).



J-Type (Jump).



R-Type (Register).



opcode	6-bit primary operation code
rd	5-bit destination register specifier
rs	5-bit source register specifier
rt	5-bit target (source/destination) register specifier or used to specify functions within the primary opcode value *REGIMM*
immediate	16-bit signed immediate used for: logical operands, arithmetic signed operands, load/store address byte offsets, PC-relative branch signed instruction displacement
instr_index	26-bit index shifted left two bits to supply the low-order 28 bits of the jump target address.
sa	5-bit shift amount
function	6-bit function field used to specify functions within the primary operation code value *SPECIAL*.

Figure A-10 CPU Instruction Formats

A 8 CPU Instruction Encoding

This section describes the encoding of user-level, i.e. non-privileged, CPU instructions for the four levels of the MIPS architecture, MIPS I through MIPS IV. Each architecture level includes the instructions in the previous level;† MIPS IV includes all instructions in MIPS I, MIPS II, and MIPS III. This section presents eight different views of the instruction encoding.

- Separate encoding tables for each architecture level.
- A MIPS IV encoding table showing the architecture level at which each opcode was originally defined and subsequently modified (if modified).
- Separate encoding tables for each architecture revision showing the changes made during that revision.

A 8.1 Instruction Decode

Instruction field names are printed in **bold** in this section.

The primary **opcode** field is decoded first. Most **opcode** values completely specify an instruction that has an immediate value or offset. **Opcode** values that do not specify an instruction specify an instruction class. Instructions within a class are further specified by values in other fields. The **opcode** values *SPECIAL* and *REGIMM* specify instruction classes. The *COP0*, *COP1*, *COP2*, *COP3*, and *COP1X* instruction classes are not CPU instructions; they are discussed in section A 8.3.

A 8.1.1 *SPECIAL* Instruction Class

The **opcode**=*SPECIAL* instruction class encodes 3-register computational instructions, jump register, and some special purpose instructions. The class is further decoded by examining the **format** field. The **format** values fully specify the CPU instructions; the *MOVCI* instruction class is not a CPU instruction class.

A 8.1.2 *REGIMM* Instruction Class

The **opcode**=*REGIMM* instruction class encodes conditional branch and trap immediate instructions. The class is further decode, and the instructions fully specified, by examining the **rt** field.

A 8.2 Instruction Subsets of MIPS III and MIPS IV Processors.

MIPS III processors, such as the R4000, R4200, R4300, R4400, and R4600, have a processor mode in which only the MIPS II instructions are valid. The MIPS II encoding table describes the MIPS II-only mode except that the Coprocessor 3 instructions (COP3, LWC3, SWC3, LDC3, SDC3) are not available and cause a Reserved Instruction exception.

† An exception to this rule is that the reserved, but never implemented, Coprocessor 3 instructions were removed or changed to another use starting in MIPS III.

MIPS IV processors, such as the R8000 and R10000, have processor modes in which only the MIPS II or MIPS III instructions are valid. The MIPS II encoding table describes the MIPS II-only mode except that the Coprocessor 3 instructions (COP3, LWC3, SWC3, LDC3, SDC3) are not available and cause a Reserved Instruction exception. The MIPS III encoding table describes the MIPS III-only mode.

A 8.3 Non-CPU Instructions in the Tables

The encoding tables show all values for the field they describe and by doing this they include some entries that are not user-level CPU instructions. The primary opcode table includes coprocessor instruction classes (COP0, COP1, COP2, COP3/COP1X) and coprocessor load/store instructions (LWCx, SWCx, LDCx, SDCx for x=1, 2, or 3). The **opcode**=*SPECIAL* + **function**=*MOVCI* instruction class is an FPU instruction.

A 8.3.1 Coprocessor 0 - *COP0*

COP0 encodes privileged instructions for Coprocessor 0, the System Control Coprocessor. The definition of the System Control Coprocessor is processor-specific and further information on these instructions are not included in this document.

A 8.3.2 Coprocessor 1 - *COP1*, *COP1X*, *MOVCI*, and CP1 load/store.

Coprocessor 1 is the floating-point unit in the MIPS architecture. *COP1*, *COP1X*, and the (**opcode**=*SPECIAL* + **function**=*MOVCI*) instruction classes encode floating-point instructions. LWC1, SWC1, LDC1, and SDC1 are floating-point loads and stores. The FPU instruction encoding is documented in section B.12.

A 8.3.3 Coprocessor 2 - *COP2* and CP2 load/store.

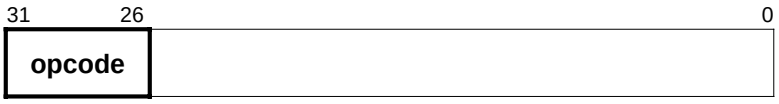
Coprocessor 2 is optional and implementation-specific. No standard processor from MIPS has implemented coprocessor 2, but MIPS' semiconductor licensees may have implemented it in a product based on one of the standard MIPS processors. At this time the standard processors are: R2000, R3000, R4000, R4200, R4300, R4400, R4600, R6000, R8000, and R10000.

A 8.3.4 Coprocessor 3 - *COP3* and CP3 load/store.

Coprocessor 3 is optional and implementation-specific in the MIPS I and MIPS II architecture levels. It was removed from MIPS III and later architecture levels. Note that in MIPS IV the *COP3* primary opcode was reused for the *COP1X* instruction class. No standard processor from MIPS has implemented coprocessor 2, but MIPS' semiconductor licensees may have implemented it in a product based on one of the standard MIPS processors. At this time the standard processors are: R2000, R3000, R4000, R4200, R4300, R4400, R4600, R6000, R8000, and R10000.

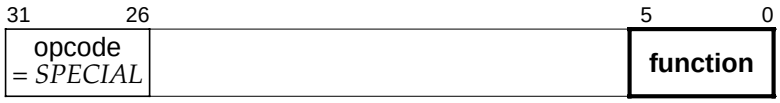
A 8.4 CPU Instruction Encoding (MIPS I)

Table A-37 CPU Instruction Encoding - MIPS I Architecture



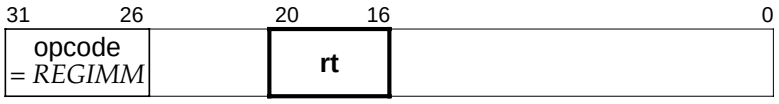
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000	*SPECIAL* δ	*REGIMM* δ	J	JAL	BEQ	BNE	BLEZ	BGTZ
1 001	ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
2 010	*COP0* δ,π	*COP1* δ,π	*COP2* δ,π	*COP3* δ,π,κ	*	*	*	*
3 011	*	*	*	*	*	*	*	*
4 100	LB	LH	LWL	LW	LBU	LHU	LWR	*
5 101	SB	SH	SWL	SW	*	*	SWR	*
6 110	*	LWC1 π	LWC2 π	LWC3 π,κ	*	*	*	*
7 111	*	SWC1 π	SWC2 π	SWC3 π,κ	*	*	*	*



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000	SLL	*	SRL	SRA	SLLV	*	SRLV	SRAV
1 001	JR	JALR	*	*	SYSCALL	BREAK	*	*
2 010	MFHI	MTHI	MFLO	MTLO	*	*	*	*
3 011	MULT	MULTU	DIV	DIVU	*	*	*	*
4 100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
5 101	*	*	SLT	SLTU	*	*	*	*
6 110	*	*	*	*	*	*	*	*
7 111	*	*	*	*	*	*	*	*

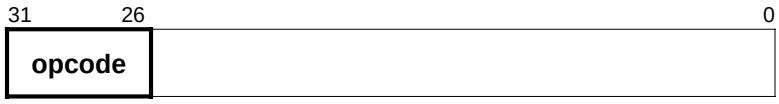


rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 00	BLTZ	BGEZ	†	†	†	†	†	†
1 01	†	†	†	†	†	†	†	†
2 10	BLTZAL	BGEZAL	†	†	†	†	†	†
3 11	†	†	†	†	†	†	†	†

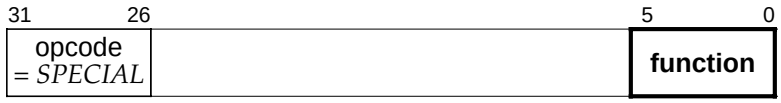
A 8.5 CPU Instruction Encoding (MIPS II)

Table A-38 CPU Instruction Encoding - MIPS II Architecture



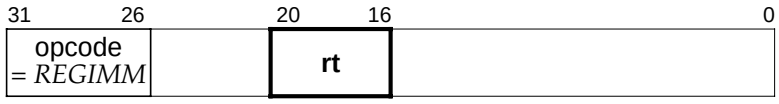
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	*SPECIAL* δ	*REGIMM* δ	J	JAL	BEQ	BNE	BLEZ	BGTZ
1	001	ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
2	010	*COP0* δ,π	*COP1* δ,π	*COP2* δ,π	*COP3* δ,π,κ	BEQL	BNEL	BLEZL	BGTZL
3	011	*	*	*	*	*	*	*	*
4	100	LB	LH	LWL	LW	LBU	LHU	LWR	*
5	101	SB	SH	SWL	SW	*	*	SWR	ρ
6	110	LL	LWC1 π	LWC2 π	LWC3 π,κ	*	LDC1 π	LDC2 π	LDC3 π,κ
7	111	SC	SWC1 π	SWC2 π	SWC3 π,κ	*	SDC1 π	SDC2 π	SDC3 π,κ



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	SLL	*	SRL	SRA	SLLV	*	SRLV	SRAV
1	001	JR	JALR	*	*	SYSCALL	BREAK	*	SYNC
2	010	MFHI	MTHI	MFLO	MTLO	*	*	*	*
3	011	MULT	MULTU	DIV	DIVU	*	*	*	*
4	100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
5	101	*	*	SLT	SLTU	*	*	*	*
6	110	TGE	TGEU	TLT	TLTU	TEQ	*	TNE	*
7	111	*	*	*	*	*	*	*	*

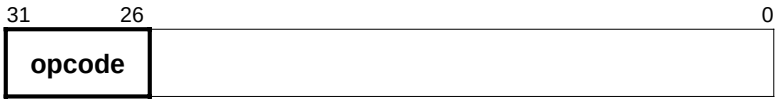


rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	00	BLTZ	BGEZ	BLTZL	BGEZL	*	*	*	*
1	01	TGEI	TGEIU	TLTI	TLTIU	TEQI	*	TNEI	*
2	10	BLTZAL	BGEZAL	BLTZALL	BGEZALL	*	*	*	*
3	11	*	*	*	*	*	*	*	*

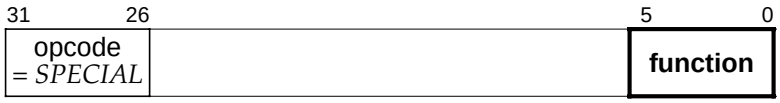
A 8.6 CPU Instruction Encoding (MIPS III)

Table A-39 CPU Instruction Encoding - MIPS III Architecture



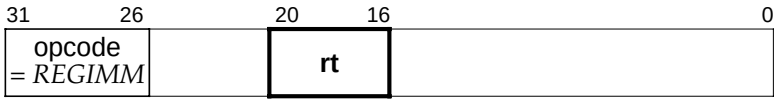
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	*SPECIAL* δ	*REGIMM* δ	J	JAL	BEQ	BNE	BLEZ	BGTZ
1	001	ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
2	010	*COP0* δ,π	*COP1* δ,π	*COP2* δ,π	*	BEQL	BNEL	BLEZL	BGTZL
3	011	DADDI	DADDIU	LDL	LDR	*	*	*	*
4	100	LB	LH	LWL	LW	LBU	LHU	LWR	LWU
5	101	SB	SH	SWL	SW	SDL	SDR	SWR	ρ
6	110	LL	LWC1 π	LWC2 π	*	LLD	LDC1 π	LDC2 π	LD
7	111	SC	SWC1 π	SWC2 π	*	SCD	SDC1 π	SDC2 π	SD



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	SLL	*	SRL	SRA	SLLV	*	SRLV	SRAV
1	001	JR	JALR	*	*	SYSCALL	BREAK	*	SYNC
2	010	MFHI	MTHI	MFLO	MTLO	DSLLV	*	DSRLV	DSRAV
3	011	MULT	MULTU	DIV	DIVU	DMULT	DMULTU	DDIV	DDIVU
4	100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
5	101	*	*	SLT	SLTU	DADD	DADDU	DSUB	DSUBU
6	110	TGE	TGEU	TLT	TLTU	TEQ	*	TNE	*
7	111	DSLL	*	DSRL	DSRA	DSLL32	*	DSRL32	DSRA32

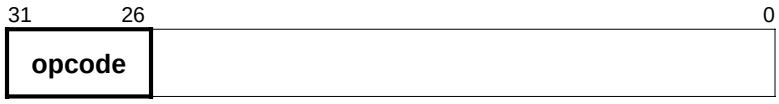


rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	00	BLTZ	BGEZ	BLTZL	BGEZL	*	*	*	*
1	01	TGEI	TGEIU	TLTI	TLTIU	TEQI	*	TNEI	*
2	10	BLTZAL	BGEZAL	BLTZALL	BGEZALL	*	*	*	*
3	11	*	*	*	*	*	*	*	*

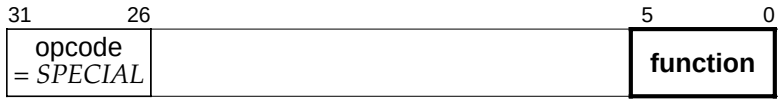
A 8.7 CPU Instruction Encoding (MIPS IV)

Table A-40 CPU Instruction Encoding - MIPS IV Architecture



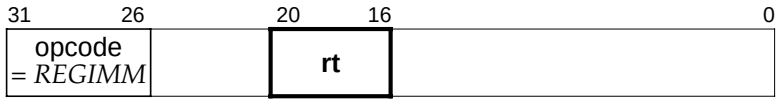
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	*SPECIAL* δ	*REGIMM* δ	J	JAL	BEQ	BNE	BLEZ	BGTZ
1	001	ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
2	010	*COP0* δ,π	*COP1* δ,π	*COP2* δ,π	*COP1X* δ,π	BEQL	BNEL	BLEZL	BGTZL
3	011	DADDI	DADDIU	LDL	LDR		*	*	*
4	100	LB	LH	LWL	LW	LBU	LHU	LWR	LWU
5	101	SB	SH	SWL	SW	SDL	SDR	SWR	ρ
6	110	LL	LWC1 π	LWC2 π	PREF	LLD	LDC1 π	LDC2 π	LD
7	111	SC	SWC1 π	SWC2 π	*	SCD	SDC1 π	SDC2 π	SD



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	SLL	*MOVCI* δ,μ	SRL	SRA	SLLV	*	SRLV	SRAV
1	001	JR	JALR	MOVZ	MOVN	SYSCALL	BREAK	*	SYNC
2	010	MFHI	MTHI	MFLO	MTLO	DSLLV	*	DSRLV	DSRAV
3	011	MULT	MULTU	DIV	DIVU	DMULT	DMULTU	DDIV	DDIVU
4	100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
5	101	*	*	SLT	SLTU	DADD	DADDU	DSUB	DSUBU
6	110	TGE	TGEU	TLT	TLTU	TEQ	*	TNE	*
7	111	DSLL	*	DSRL	DSRA	DSLL32	*	DSRL32	DSRA32

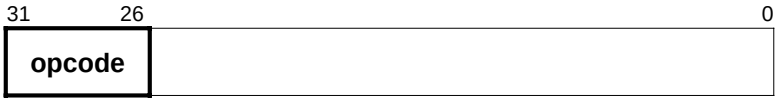


rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	00	BLTZ	BGEZ	BLTZL	BGEZL	*	*	*	*
1	01	TGEI	TGEIU	TLTI	TLTIU	TEQI	*	TNEI	*
2	10	BLTZAL	BGEZAL	BLTZALL	BGEZALL	*	*	*	*
3	11	*	*	*	*	*	*	*	*

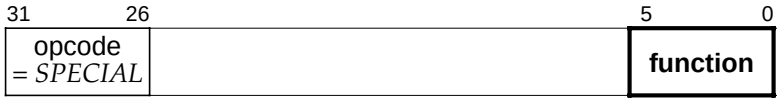
Table A-41 Architecture Level in Which CPU Instructions are Defined or Extended.

The architecture level in which each MIPS IVencoding was defined is indicated by a subscript 1, 2, 3, or 4 (for architecture level I, II, III, or IV). If an instruction or instruction class was later extended, the extending level is indicated after the defining level.



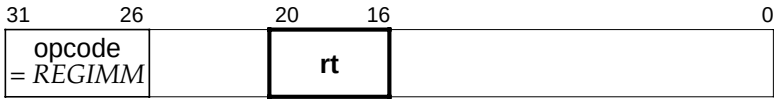
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	*SPECIAL* $_{1-4}$	*REGIMM* $_{1,2}$	J $_1$	JAL $_1$	BEQ $_1$	BNE $_1$	BLEZ $_1$	BGTZ $_1$
1	001	ADDI $_1$	ADDIU $_1$	SLTI $_1$	SLTIU $_1$	ANDI $_1$	ORI $_1$	XORI $_1$	LUI $_1$
2	010	*COP0* $_1$	*COP1* $_{1,2,3,4}$	*COP2* $_1$	*COP1X* $_4$	BEQL $_2$	BNEL $_2$	BLEZL $_2$	BGTZL $_2$
3	011	DADDI $_3$	DADDIU $_3$	LDL $_3$	LDR $_3$	* $_1$	* $_1$	* $_1$	* $_1$
4	100	LB $_1$	LH $_1$	LWL $_1$	LW $_1$	LBU $_1$	LHU $_1$	LWR $_1$	LWU $_3$
5	101	SB $_1$	SH $_1$	SWL $_1$	SW $_1$	SDL $_3$	SDR $_3$	SWR $_1$	ρ $_2$
6	110	LL $_2$	LWC1 $_1$	LWC2 $_1$	PREF $_4$	LLD $_3$	LDC1 $_2$	LDC2 $_2$	LD $_3$
7	111	SC $_2$	SWC1 $_1$	SWC2 $_1$	*$_3$	SCD $_3$	SDC1 $_2$	SDC2 $_2$	SD $_3$



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	000	SLL $_1$	*MOVCI* $_4$	SRL $_1$	SRA $_1$	SLLV $_1$	* $_1$	SRLV $_1$	SRAV $_1$
1	001	JR $_1$	JALR $_1$	MOVZ $_4$	MOVN $_4$	SYSCALL $_1$	BREAK $_1$	* $_1$	SYNC $_2$
2	010	MFHI $_1$	MTHI $_1$	MFLO $_1$	MTLO $_1$	DSLLV $_3$	* $_1$	DSRLV $_3$	DSRAV $_3$
3	011	MULT $_1$	MULTU $_1$	DIV $_1$	DIVU $_1$	DMULT $_3$	DMULTU $_3$	DDIV $_3$	DDIVU $_3$
4	100	ADD $_1$	ADDU $_1$	SUB $_1$	SUBU $_1$	AND $_1$	OR $_1$	XOR $_1$	NOR $_1$
5	101	* $_1$	* $_1$	SLT $_1$	SLTU $_1$	DADD $_3$	DADDU $_3$	DSUB $_3$	DSUBU $_3$
6	110	TGE $_2$	TGEU $_2$	TLT $_2$	TLTU $_2$	TEQ $_2$	* $_1$	TNE $_2$	* $_1$
7	111	DSLL $_3$	* $_1$	DSRL $_3$	DSRA $_3$	DSLL32 $_3$	* $_1$	DSRL32 $_3$	DSRA32 $_3$



rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	00	BLTZ $_1$	BGEZ $_1$	BLTZL $_2$	BGEZL $_2$	* $_1$	* $_1$	* $_1$	* $_1$
1	01	TGEI $_2$	TGEIU $_2$	TLTI $_2$	TLTIU $_2$	TEQI $_2$	* $_1$	TNEI $_2$	* $_1$
2	10	BLTZAL $_1$	BGEZAL $_1$	BLTZALL $_2$	BGEZALL $_2$	* $_1$	* $_1$	* $_1$	* $_1$
3	11	* $_1$	* $_1$	* $_1$	* $_1$	* $_1$	* $_1$	* $_1$	* $_1$

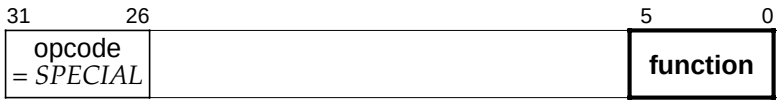
A 8.8 CPU Instruction Encoding Changes (MIPS II)

Table A-42 CPU Instruction Encoding Changes - MIPS II Revision.

An instruction encoding is shown if the instruction is added in this revision.

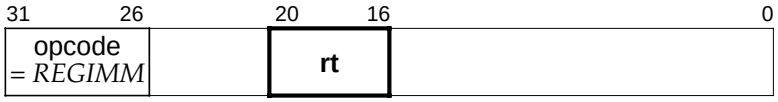
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000								
1 001								
2 010					BEQL	BNEL	BLEZL	BGTZL
3 011								
4 100								
5 101								ρ
6 110	LL					LDC1 π	LDC2 π	LDC3 π
7 111	SC					SDC1 π	SDC2 π	SDC3 π



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000								
1 001								SYNC
2 010								
3 011								
4 100								
5 101								
6 110	TGE	TGEU	TLT	TLTU	TEQ		TNE	
7 111								



rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 00			BLTZL	BGEZL				
1 01	TGEI	TGEIU	TLTI	TLTIU	TEQI		TNEI	
2 10			BLTZALL	BGEZALL				
3 11								

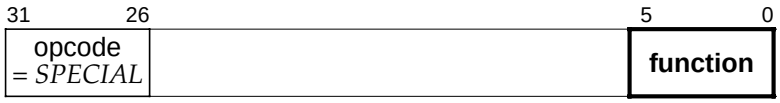
A 8.9 CPU Instruction Encoding Changes (MIPS III)

Table A-43 CPU Instruction Encoding Changes - MIPS III Revision.

An instruction encoding is shown if the instruction is added or modified in this revision.

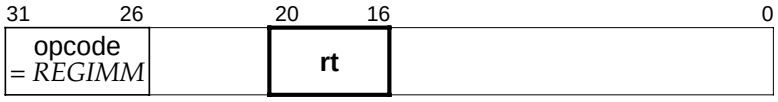
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000								
1 001								
2 010				* (was *COP3*)				
3 011	DADDI	DADDIU	LDL	LDR				
4 100								LWU
5 101					SDL	SDR		
6 110				* (was LWC3)	LLD			LD (was LDC3)
7 111				* (was SWC3)	SCD			SD (was SDC3)



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000								
1 001								
2 010					DSLLV		DSRLV	DSRAV
3 011					DMULT	DMULTU	DDIV	DDIVU
4 100								
5 101					DADD	DADDU	DSUB	DSUBU
6 110								
7 111	DSLL		DSRL	DSRA	DSLL32		DSRL32	DSRA32



rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 00								
1 01								
2 10								
3 11								

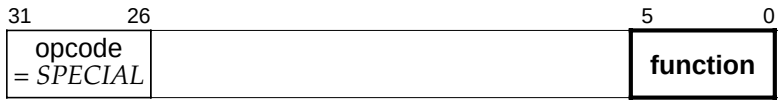
A 8.10 CPU Instruction Encoding Changes (MIPS IV)

Table A-44 CPU Instruction Encoding Changes - MIPS IV Revision.

An instruction encoding is shown if the instruction is added or modified in this revision.

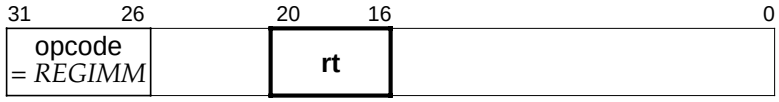
opcode bits 28..26 Instructions encoded by **opcode** field.

bits 31..29	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000								
1 001								
2 010				*COP1X* δ,π				
3 011								
4 100								
5 101								
6 110				PREF				
7 111								



function bits 2..0 Instructions encoded by **function** field when opcode field = SPECIAL.

bits 5..3	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000		*MOVCI* δ,μ						
1 001			MOVZ	MOVN				
2 010								
3 011								
4 100								
5 101								
6 110								
7 111								



rt bits 18..16 Instructions encoded by the **rt** field when opcode field = REGIMM.

bits 20..19	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 00								
1 01								
2 10								
3 11								

Key to notes in CPU instruction encoding tables:

***(asterisk)** This opcode is reserved for future use. An attempt to execute it causes a Reserved Instruction exception.

(cross) This opcode is reserved for future use. An attempt to execute it produces an undefined result. The result may be a Reserved Instruction exception but this is not guaranteed.

δ(delta) (also *italic* opcode name) This opcode indicates an instruction class. The instruction word must be further decoded by examing additional tables that show values for another instruction field.

π(pi) This opcode is a coprocessor operation, not a CPU operation. If the processor state does not allow access to the specified coprocessor, the instruction causes a Coprocessor Unusable exception. It is included in the table because it uses a primary opcode in the instruction encoding map.

κ(kappa) This opcode is removed in a later revision of the architecture. If a MIPS III or MIPS IV processor is operated in MIPS II-only mode this opcode will cause a Reserved Instruction exception.

μ(mu) This opcode indicates a class of coprocessor 1 instructions. If the processor state does not allow access to coprocessor 1, the opcode causes a Coprocessor Unusable exception. It is included in the table because the encoding uses a location in what is otherwise a CPU instruction encoding map. Further encoding information for this instruction class is in the FPU Instruction Encoding tables.

ρ(rho) This opcode is reserved for Coprocessor 0 (System Control Coprocessor) instructions that require base+offset addressing. If the instruction is used for COP0 in an implementation, an attempt to execute it without Coprocessor 0 access privilege will cause a Coprocessor Unusable exception. If the instruction is not used in an implementation, it will cause a Reserved Instruction exception.