Instruction encodings

32 bit encodings

Abbreviations used

cond : a three bit field specifying execution condition (See table xx at the end)
creg : a three bit field specifying one of the eight condition registers
cp_sreg/sr : a five bit field selecting a coprocessor source register
cp_dreg/dr : a five bit field selecting a coprocessor destination register
dreg : a five bit field selecting a destination register
dr : a three bit field selecting a destination register
imm, imm1, imm2
: an immediate constant
imml : left part of an immediate constant
immr : right part of an immediate constant
msb : with lui & lli instructions the most significant bit of a 16 bit immediate.
*sreg*i : a five bit field selecting a source register (i = ‘ ‘, 1, 2)
*sr*i : a three bit field selecting a source register(i = ‘ ‘, 1, 2)
cex : A bit specifying if conditional execution of an instruction is enabled or not. If low the cond and creg fields can be used to represent a longer immediate.
xxx : A bit field which has no meaning. The assembler can decide what bit patterns to use.

About notation

The instruction word is presented as adjacent bitfields with symbolic names. The numbers below the name of a bitfield are the range of bit indexes reserved by the field. One can see the exact position and the length of the bitfiled from the indexes. The order of significance is from left to right, that is bit index 31 corresponds the msb of the word and index 0 corresponds to lsb of the word. Conditionally executable instructions, which have an immediate as the second operand, have two encodings. The first one is with conditional execution (*cex* = 1) and the second is without conditional execution (*cex* = 0). The latter encoding expands the immediate constant over the *cond* and *creg* fields.

The first field of each instruction is instruction identifier (iid). From iid the processor decodes following: What operation to perform, what kind of operands to use and if there’s any constraints on the result or usage of instruction. The iid could also be called opcode even though it strictly speaking isn’t plain operation code. Some instructions use the same iid in which case other fields are used to distinguish between them.

If an assembler does not conform to this encoding the results are unpredictable.
The reader should refer to the chapter ‘Instruction specifications’ for information about the description of each instruction.

(cond, creg) **add** dreg, sreg1, sreg2

000001	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **addi** dreg, sreg1, imm

101101	1	creg	cond	imm	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0
101101	0	imm			sreg1	dreg
31...26	25	24...10			9...5	4...0

(cond, creg) **addiu** dreg, sreg1, imm

101000	1	creg	cond	imm	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0
101000	0	imm			sreg1	dreg
31...26	25	24...10			9...5	4...0

(cond, creg) **addu** dreg, sreg1, sreg2

000000	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **and** dreg, sreg1, sreg2

000010	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **andi** dreg, sreg1, imm

101001	1	creg	cond	imm	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0
101001	0	imm			sreg1	dreg
31...26	25	24...10			9...5	4...0

bc creg, imm

100000	1	creg	imm
31...26	25	24...22	21...0

begt creg, imm

100001	1	creg	imm
31...26	25	24...22	21...0

belt creg, imm

100010	1	creg	imm
31...26	25	24...22	21...0

beq creg, imm

100011	1	creg	imm
31...26	25	24...22	21...0

bgt creg, imm

100100	1	creg	imm
31...26	25	24...22	21...0

blt creg, imm

100101	1	creg	imm
31...26	25	24...22	21...0

bne creg, imm

100110	1	creg	imm
31...26	25	24...22	21...0

bnc creg, imm

100111	1	creg	imm
31...26	25	24...22	21...0

chrs imm

110011	0	xxx	imm	xxx
31...26	25	24...12	11...10	9...0

cmp creg, sreg1, sreg2

011001	0	creg	xxx	sreg2	sreg1	xxx
31...26	25	24...22	21...15	14...10	9...5	4...0

cmpi creg, sreg1, imm

110111	0	creg	immr	sreg1	imml
31...26	25	24...22	21...10	9...5	4...0

(cond, creg) **conb** dreg, sreg1, sreg2

000011	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **conh** dreg, sreg2, sreg1

000100	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

cop imm1, imm2

111100	imm1	imm2
31...26	25...24	23...0

di

010101	0	xxx
31...26	25	24...0

ei

010110	0	xxx
31...26	25	24...0

(cond, creg) **exb** dreg, sreg1, imm

110000	cex	creg	cond	xxx	imm	sreg1	dreg
31...26	25	24...22	21...19	18...12	11...10	9...5	4...0

(cond, creg) **exbf** dreg, sreg1, sreg2

011010	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

exbfi dreg, sreg1, imm1, imm2

111101	0	xxx	imm1	imm2	sreg1	dreg
31...26	25	24...21	20...15	14...10	9...5	4...0

(cond, creg) **exh** dreg, sreg1, imm

110001	cex	creg	cond	xxx	imm	sreg1	dreg
31...26	25	24...22	21...19	18...11	10	9...5	4...0

jal imm

111001	0	imm
31...26	25	24...0

(cond, creg) **jalr** sreg1

110101	cex	creg	cond	xxx	sreg1	11111
31...26	25	24...22	21...19	18...10	9...5	4...0

jmp imm

111000	0	imm
31...26	25	24...0

(cond, creg) **jmpr** sreg1

011011	cex	creg	cond	xxx	sreg1	xxx
31...26	25	24...22	21...19	18...10	9...5	4...0

(cond, creg) **ld** dreg, sreg1, imm

110010	1	creg	cond	imm	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0
110010	0	imm			sreg1	dreg
31...26	25	24...10			9...5	4...0

lli dreg, imm

111110	0	imm	msb	xxx	dreg
31...26	25	24...10	9	8...5	4...0

lui dreg, imm

111111	0	imm	msb	xxx	dreg
31...26	25	24...10	9	8...5	4...0

(cond, creg) **mov** dreg, sreg1

010011	cex	creg	cond	xxx	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0

(cond, creg) **movfc** imm, dreg, cp_sreg

101100	cex	creg	cond	xxx	cp_sreg	imm	xxx	dreg
31...26	25	24...22	21...19	18...17	16...12	11...10	9...5	4...0

(cond, creg) **movtc** imm, cp_dreg, sreg1

110110	cex	creg	cond	xxx	cp_dreg	imm	sreg1	xxx
31...26	25	24...22	21...19	18...17	16...12	11...10	9...5	4...0

(cond, creg) **mulhi** dreg

011101	cex	creg	cond	xxx	dreg
31...26	25	24...22	21...19	18...5	4...0

(cond, creg) **muli** dreg, sreg1, imm

101110	1	creg	cond	imm	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0
101110	0	imm			sreg1	dreg
31...26	25	24...10			9...5	4...0

(cond, creg) **muls** dreg, sreg1, sreg2

000101	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **muls_16** dreg, sreg1, sreg2

001000	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **mulu** dreg, sreg1, sreg2

000110	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **mulu_16** dreg, sreg1, sreg2

001001	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **mulus** dreg, sreg1, sreg2

000111	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **mulus_16** dreg, sreg1, sreg2

001010	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

nop

111010	0	xxx	111010	xxx
31...26	25	24...16	15...10	9...0

(cond, creg) **not** dreg, sreg1

010100	cex	creg	cond	xxx	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0

(cond, creg) **or** dreg, sreg1, sreg2

001011	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **ori** dreg, sreg1, imm

101010	1	creg	cond	imm	sreg1	dreg
31...26	25	24...22	21...19	18...10	9...5	4...0
101010	0	imm			sreg1	dreg
31...26	25	24...10			9...5	4...0

rcon sreg1

011110	0	xxx	sreg1	xxx
31...26	25	24...10	9...5	4...0

reti

010111	0	000	xxx
31...26	25	24...22	21...0

retu

011111	0	xxx	11111	xxx
31...26	25	24...10	9...5	4...0

(cond, creg) **scall**

111011	cex	creg	cond	xxx	11111
31...26	25	24...22	21...19	18...5	4...0

scon dreg

011100	0	xxx	dreg
31...26	25	24...5	4...0

(cond, creg) **sext** dreg, sreg1, sreg2

001100	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **sexti** dreg, sreg1, imm

101011	1	creg	cond	xxx	imm	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0
101011	0	xxx		imm		sreg1	dreg
31...26	25	24...15		14...10		9...5	4...0

(cond, creg) **sll** dreg, sreg1, sreg2

001101	cex	creg	cond	1	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18	17...15	14...10	9...5	4...0

(cond, creg) **slli** dreg, sreg1, imm

001101	cex	creg	cond	0	xxx	imm	sreg1	dreg
31...26	25	24...22	21...19	18	17...16	15...10	9...5	4...0

(cond, creg) **sra** dreg, sreg1, sreg2

001110	cex	creg	cond	1	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18	17...15	14...10	9...5	4...0

(cond, creg) **srai** dreg, sreg1, imm

001110	cex	creg	cond	0	xxx	imm	sreg1	dreg
31...26	25	24...22	21...19	18	17...16	15...10	9...5	4...0

(cond, creg) **srl** dreg, sreg1, sreg2

001111	cex	creg	cond	1	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18	17...15	14...10	9...5	4...0

(cond, creg) **srli** dreg, sreg1, imm

001111	cex	creg	cond	0	xxx	imm	sreg1	dreg
31...26	25	24...22	21...19	18	17...16	15...10	9...5	4...0

(cond, creg) **st** sreg2, sreg1, imm

110100	1	creg	cond	immr	sreg2	sreg1	imml
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0
110100	0	immr			sreg2	sreg1	imml
31...26	25	24...15			14...10	9...5	4...0

(cond, creg) **sub** dreg, sreg1, sreg2

010000	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

(cond, creg) **subu** dreg, sreg1, sreg2

010001	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

swm imm

101111	0	xxx	imm	xxx
31...26	25	24...16	15...10	9...0

(cond, creg) **trap** imm

011000	cex	creg	cond	xxx	imm	xxx
31...26	25	24...22	21...19	18...15	14...10	9...0

(cond, creg) **xor** dreg, sreg1, sreg2

010010	cex	creg	cond	xxx	sreg2	sreg1	dreg
31...26	25	24...22	21...19	18...15	14...10	9...5	4...0

16 bit encodings

Notes about 16 bit encoding

Almost all instructions have both a 16 bit and a 32 bit version. Instructions which are valid only in 32 bit mode are: cop, exbfi, lli and lui. Conditional execution is not allowed in 16 bit mode, so fields *creg*, *cond* and *cex* are not needed. Also source and destination register fields and immediate fields are shorter. See Chapter ‘Operating modes’ for more information.

add dr, sr

000001	sr	xxx	dr
15...10	9...7	6...3	2...0

addi dr, imm

101101	imm	dr
15...10	9...3	2...0

addiu dr, imm

101000	imm	dr
15...10	9...3	2...0

addu dr, sr

000000	sr	xxx	dr
15...10	9...7	6...3	2...0

and dr, sr

000010	sr	xxx	dr
15...10	9...7	6...3	2...0

andi dr, imm

101001	imm	dr
15...10	9...3	2...0

bc imm

100000	imm
15...10	9...0

begt, imm

100001	imm
15...10	9...0

belt, imm

100010	imm
15...10	9...0

beq, imm

100011	imm
15...10	9...0

bgt, imm

100100	imm
15...10	9...0

blt, imm

100101	imm
15...10	9...0

bne, imm

100110	imm
15...10	9...0

bnc imm

100111	imm
15...10	9...0

chrs imm

110011	xxx	imm	xxx
15...10	9...5	4...3	2...0

cmp sr1, sr2

011001	sr1	xxx	sr2
15...10	9...7	6...3	2...0

cmpi sr, imm

110111	sr	imm
15...10	9...7	6...0

conb dr, sr

000011	sr	xxx	dr
15...10	9...7	6...3	2...0

conh dr, sr

000100	sr	xxx	dr
15...10	9...7	6...3	2...0

di

010101	xxx

15...10	9...0

ei

010110	xxx
15...10	9...0

exb dr, sr, imm

110000	sr	xxx	imm	dr
15...10	9...7	6...5	4...3	2...0

exbf dr, sr

011010	sr	xxx	dr
15...10	9...7	6...3	2...0

exh dr, sr, imm

110001	sr	xxx	imm	dr
15...10	9...7	6...4	3	2...0

jal imm

111001	imm
15...10	9...0

jalr sr

110101	sr	xxx	111
15...10	9...7	6...3	2...0

jmp imm

111000	imm
15...10	9...0

jmpr sr

011011	sr	xxx
15...10	9...7	6...0

ld dr, sr, imm

110010	sr	imm	dr
15...10	9...7	6...3	2...0

mov dr, sr

010011	sr	xxx	dr
15...10	9...7	6...3	2...0

movfc imm, dr, cp_sr

101100	cp_sr	imm	dr
15...10	9...5	4..3	2...0

movtc imm, c_dr, sr

110110	sr	cp_dr	imm
15...10	9...7	6...2	1...0

mulhi dr

011101	xxx	xxx	dr
15...10	9...7	6...3	2...0

muli dr, imm

101110	imm	dr
15...10	9...3	2...0

muls dr, sr

000101	sr	xxx	dr
15...10	9...7	6...3	2...0

muls_16 dr, sr

001000	sr	xxx	dr
15...10	9...7	6...3	2...0

mulu dr, sr

000110	sr	xxx	dr
15...10	9...7	6...3	2...0

mulu_16 dr, sr

001001	sr	xxx	dr
15...10	9...7	6...3	2...0

mulus dr, sr

000111	sr	xxx	dr
15...10	9...7	6...3	2...0

mulus_16 dr, sr

001010	sr	xxx	dr
15...10	9...7	6...3	2...0

nop

111010	xxx
15...10	9...0

not dr, sr

010100	sr	xxx	dr
15...10	9...7	6...3	2...0

or dr, sr

001011	sr	xxx	dr
15...10	9...7	6...3	2...0

ori dr, imm

101010	imm	dr
15...10	9...3	2...0

rcon sr

011110	sr	xxx
15...10	9...7	6...0

reti

010111	xxx
15...10	9...0

retu

011111	xxx
15...10	9...0

scall

111011	xxx	111
15...10	9...3	2...0

scon dr

011100	xxx	dr
15...10	9...3	2...0

sext dr, sr

001100	sr	xxx	dr
15...10	9...7	6...3	2...0

sexti dr, imm

101011	xxx	imm	dr
15...10	9...8	7...3	2...0

sll dr, sr

001101	sr	xxx	1	dr
15...10	9...7	6...4	3	2...0

slli dr, imm

001101	imm	0	dr
15...10	9...4	3	2...0

sra dr, sr

001110	sr	xxx	1	dr
15...10	9...7	6...4	3	2...0

srai dr, imm

001110	imm	0	dr
15...10	9...4	3	2...0

srl dr, sr

001111	sr	xxx	1	dr
15...10	9...7	6...4	3	2...0

srli dr, imm

001111	imm	0	dr
15...10	9...4	3	2...0

st sr2, sr1, imm

110100	sr1	imm	sr2
15...10	9...7	6...3	2...0

sub dr, sr

010000	sr	xxx	dr
15...10	9...7	6...3	2...0

subu dr, sr

010001	sr	xxx	dr
15...10	9...7	6...3	2...0

swm imm

101111	x	imm	xxx
15...10	9	8...3	2...0

trap imm

011000	xxx	imm	xxx
15...10	9...8	7...3	2...0

xor dr, sr

010010	sr	xxx	dr
15...10	9...7	6...3	2...0

mnemonic	condition	explanation	code	flags
c	carry	Carry out of MSB	000	C = 1
eq	=	equal	011	Z = 1
gt	>	greater than	100	Z = 0 & N = 0
lt	<	less than	101	Z = 0 & N = 1
ne	≠	not equal	110	Z = 0
elt	≤	equal or less than	010	Z = 1 or N = 1
egt	≥	equal or greater than	001	Z = 1 or N = 0
nc	!carry	No carry-out	111	C = 0

Table xx, Condition codes