ECE/CS 250 EXAM REFERENCE SHEET

(You may keep or discard this sheet)

MIPS info

MIPS32® Instruction Set Quick Reference

RD — DESTINATION REGISTER
RS, RT — SOURCE OPERAND REGISTERS
RA — RETURN ADDRESS REGISTER (R31)
PC — PROGRAM COUNTER
ACC — 64-BIT ACCUMULATOR
LO, HI — ACCUMULATOR LOW ($ACC_{31:0}$) AND HIGH ($ACC_{63:32}$) PARTS
— SIGNED OPERAND OR SIGN EXTENSION
— UNSIGNED OPERAND OR ZERO EXTENSION
:: — CONCATENATION OF BIT FIELDS
R2 — MIPS32 RELEASE 2 INSTRUCTION
DOTTED — ASSEMBLER PSEUDO-INSTRUCTION

PLEASE REFER TO *"MIPS32 ARCHITECTURE FOR PROGRAMMERS VOLUME II: THE MIPS32 INSTRUCTION SET"* FOR COMPLETE INSTRUCTION SET INFORMATION.

ARITHMETIC OPERATIONS			
ADD	RD, RS, RT	RD = RS + RT	(OVERFLOW TRAP)
ADDI	RD, RS, CONST16	RD = RS + CONST16	(OVERFLOW TRAP)
ADDIU	RD, RS, CONST16	RD = RS + CONST16	
ADDU	RD, RS, RT	RD = RS + RT	
CLO	RD, RS	RD = COUNTLEADINGONES(RS)	
CLZ	RD, RS	RD = COUNTLEADINGZEROS(RS)	
LA	RD, LABEL	RD = ADDRESS(LABEL)	
LI	RD, IMM32	RD = IMM32	
LUI	RD, CONST16	RD = CONST16 << 16	
MOVE	RD, RS	RD = RS	
NEGU	RD, RS	RD = –RS	
SEB[R2]	RD, RS	$RD = RS_{7:0}$	
SEH[R2]	RD, RS	$RD = RS_{15:0}$	
SUB	RD, RS, RT	RD = RS – RT	(OVERFLOW TRAP)
SUBU	RD, RS, RT	RD = RS – RT	

SHIFT AND ROTATE OPERATIONS		
ROTR[R2]	RD, RS, BITS5	$RD = RS_{BITS5-1:0} :: RS_{31:BITS5}$
ROTRV[R2]	RD, RS, RT	$RD = RS_{RT4:0-1:0} :: RS_{31:RT4:0}$
SLL	RD, RS, SHIFT5	RD = RS << SHIFT5
SLLV	RD, RS, RT	$RD = RS << RT_{4:0}$
SRA	RD, RS, SHIFT5	RD = RS >> SHIFT5
SRAV	RD, RS, RT	$RD = RS >> RT_{4:0}$
SRL	RD, RS, SHIFT5	RD = RS >> SHIFT5
SRLV	RD, RS, RT	$RD = RS >> RT_{4:0}$

LOGICAL AND BIT-FIELD OPERATIONS		
AND	RD, RS, RT	RD = RS & RT
ANDI	RD, RS, CONST16	RD = RS & CONST16
EXT[R2]	RD, RS, P, S	$RS = RS_{P+S-1:P}$
INS[R2]	RD, RS, P, S	$RD_{P+S-1:P} = RS_{S-1:0}$
NOP		NO-OP
NOR	RD, RS, RT	RD = ~(RS | RT)
NOT	RD, RS	RD = ~RS
OR	RD, RS, RT	RD = RS | RT
ORI	RD, RS, CONST16	RD = RS | CONST16
WSBH[R2]	RD, RS	$RD = RS_{23:16} :: RS_{31:24} :: RS_{7:0} :: RS_{15:8}$
XOR	RD, RS, RT	RD = RS RT
XORI	RD, RS, CONST16	RD = RS CONST16

CONDITION TESTING AND CONDITIONAL MOVE OPERATIONS		
MOVN	RD, RS, RT	IF RT ≠ 0, RD = RS
MOVZ	RD, RS, RT	IF RT = 0, RD = RS
SLT	RD, RS, RT	RD = (RS < RT) ? 1 : 0
SLTI	RD, RS, CONST16	RD = (RS < CONST16) ? 1 : 0
SLTIU	RD, RS, CONST16	RD = (RS < CONST16) ? 1 : 0
SLTU	RD, RS, RT	RD = (RS < RT) ? 1 : 0

MULTIPLY AND DIVIDE OPERATIONS			
DIV	RS, RT	LO = RS / RT	RS MOD RT
DIVU	RS, RT	LO = RS / RT	RS MOD RT
MADD	RS, RT	ACC += RS RT	
MADDU	RS, RT	ACC += RS RT	
MSUB	RS, RT	ACC –= RS RT	
MSUBU	RS, RT	ACC –= RS RT	
MUL	RD, RS, RT	RD = RS RT	
MULT	RS, RT	ACC = RS RT	
MULTU	RS, RT	ACC = RS RT	

ACCUMULATOR ACCESS OPERATIONS		
MFHI	RD	RD = HI
MFLO	RD	RD = LO
MTHI	RS	HI = RS
MTLO	RS	LO = RS

JUMPS AND BRANCHES (NOTE: ONE DELAY SLOT)		
B	OFF18	PC += OFF18
BAL	OFF18	RA = PC + 8, PC += OFF18
BEQ	RS, RT, OFF18	IF RS = RT, PC += OFF18
BEQZ	RS, OFF18	IF RS = 0, PC += OFF18
BGEZ	RS, OFF18	IF RS ≥ 0, PC += OFF18
BGEZAL	RS, OFF18	RA = PC + 8; IF RS ≥ 0, PC += OFF18
BGTZ	RS, OFF18	IF RS > 0, PC += OFF18
BLEZ	RS, OFF18	IF RS ≤ 0, PC += OFF18
BLTZ	RS, OFF18	IF RS < 0, PC += OFF18
BLTZAL	RS, OFF18	RA = PC + 8; IF RS < 0, PC += OFF18
BNE	RS, RT, OFF18	IF RS ≠ RT, PC += OFF18
BNEZ	RS, OFF18	IF RS ≠ 0, PC += OFF18
J	ADDR28	$PC = PC_{31:28} :: ADDR28$
JAL	ADDR28	RA = PC + 8; PC = PC :: ADDR
JALR	RD, RS	RD = PC + 8; PC = RS
JR	RS	PC – RS

LOAD AND STORE OPERATIONS		
LB	RD, OFF16(RS)	RD = MEM8(RS + OFF16)
LBU	RD, OFF16(RS)	RD = MEM8(RS + OFF16)
LH	RD, OFF16(RS)	RD = MEM16(RS + OFF16)
LHU	RD, OFF16(RS)	RD = MEM16(RS + OFF16)
LW	RD, OFF16(RS)	RD = MEM32(RS + OFF16)
LWL	RD, OFF16(RS)	RD = LOADWORDLEFT(RS + OFF16)
LWR	RD, OFF16(RS)	RD – LOADWORDRIGHT(RS + OFF16)
SB	RS, OFF16(RT)	MEM8(RT + OFF16) = $RS_{7:0}$
SH	RS, OFF16(RT)	MEM16(RT + OFF16) = $RS_{15:0}$
SW	RS, OFF16(RT)	MEM32(RT + OFF16) = RS
SWL	RS, OFF16(RT)	STOREWORDLEFT(RT + OFF16 , RS)
SWR	RS, OFF16(RT)	STOREWORDRIGHT(RT + OFF16 , RS)
ULW	RD, OFF16(RS)	RD = UNALIGNED_MEM32(RS + OFF16)
USW	RS, OFF16(RT)	UNALIGNED_MEM32(RT + OFF16) = RS

ATOMIC READ-MODIFY-WRITE OPERATIONS		
LL	RD, OFF16(RS)	RD = MEM32(RS + OFF16); LINK
SC	RD, OFF16(RS)	IF ATOMIC, MEM32(RS + OFF16) = RD; RD = ATOMIC ? 1 : 0

MD00565 Revision 01.01

Powers of two

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1,024
11	2,048
12	4,096
13	8,192
14	16,384
15	32,768
16	65,536

The Intel x86 instruction set

AAA	CMOVE	CVTPS2DQ	FCMOVU	FNOP	GS	JNGE	MFENCE	MULSS	PCMPISTRM	PMULLD	PUNPCKLDQ	SETC	STOSB
AAD	CMOVG	CVTPS2PD	FCOM	FNSAVE	HADDPD	JNL	MINPD	MWAIT	PEXTRB	PMULLW	PUNPCKLQDQ	SETE	STOSD
AAM	CMOVGE	CVTPS2PI	FCOM2	FNSETPM	HADDPS	JNLE	MINPS	NEG	PEXTRD	PMULUDQ	PUNPCKLWD	SETG	STOSW
AAS	CMOVL	CVTSD2SI	FCOMI	FNSTCW	HINT_NOP	JNO	MINSD	NOP	PEXTRQ	POP	PUSH	SETGE	STR
ADC	CMOVLE	CVTSD2SS	FCOMIP	FNSTENV	HLT	JNP	MINSS	NOT	PEXTRW	POPA	PUSHA	SETL	SUB
ADD	CMOVNA	CVTSI2SD	FCOMP	FNSTSW	HSUBPD	JNS	MONITOR	OR	PHADDD	POPAD	PUSHAD	SETLE	SUBPD
ADDPD	CMOVNAE	CVTSI2SS	FCOMP3	FPATAN	HSUBPS	JNZ	MOV	ORPD	PHADDSW	POPCNT	PUSHF	SETNA	SUBPS
ADDPS	CMOVNB	CVTSS2SD	FCOMP5	FPREM	ICEBP	JO	MOVAPD	ORPS	PHADDW	POPF	PUSHFD	SETNAE	SUBSD
ADDSD	CMOVNBE	CVTSS2SI	FCOMPP	FPREM1	IDIV	JP	MOVAPS	OUT	PHMINPOSUW	POPFD	PXOR	SETNB	SUBSS
ADDSS	CMOVNC	CVTTPS2DQ	FCOS	FPTAN	IMUL	JPE	MOVBE	OUTS	PHSUBD	POR	RCL	SETNBE	SYSENTER
ADDSUBPD	CMOVNE	CVTTPD2PI	FDECSTP	FRNDINT	IN	JPO	MOVD	OUTSB	PHSUBSW	PREFETCHNTA	RCPPS	SETNC	SYSEXIT
ADDSUBPS	CMOVNG	CVTTPS2DQ	FDIV	FRSTOR	INC	JS	MOVDDUP	OUTSD	PHSUBW	PREFETCHT0	RCPSS	SETNE	TEST
ADX	CMOVNGE	CVTTPS2PI	FDIVP	FS	INS	JZ	MOVDQ2Q	OUTSW	PINSRB	PREFETCHT1	RCR	SETNG	UCOMISD
AMX	CMOVNL	CVTTSD2SI	FDIVR	FSAVE	INSB	LAHF	MOVDQA	PABSB	PINSRD	PREFETCHT2	RDMSR	SETNGE	UCOMISS
AND	CMOVNLE	CVTTSS2SI	FDIVRP	FSCALE	INSD	LAR	MOVDQU	PABSD	PINSRQ	PSADBW	RDPMC	SETNL	UD
ANDNPD	CMOVNO	CWD	FFREE	FSIN	INSERTPS	LDDQU	MOVHLPS	PABSW	PINSRW	PSHUFB	RDTSC	SETNLE	UD2
ANDNPS	CMOVNP	CWDE	FFREEP	FSINCOS	INSW	LDMXCSR	MOVHPD	PACKSSDW	PMADDUBSW	PSHUFD	RDTSCP	SETNO	UNPCKHPD
ANDPD	CMOVNS	DAA	FIADD	FSQRT	INT	LDS	MOVHPS	PACKSSWB	PMAXSD	PSHUFHW	REP	SETNP	UNPCKHPS
ANDPS	CMOVNZ	DAS	FICOM	FST	INT1	LEA	MOVLHPS	PACKUSDW	PMAXSB	PSHUFLW	REPE	SETNS	UNPCKLPD
ARPL	CMOVO	DEC	FICOMP	FSTCW	INTO	LEAVE	MOVLPD	PACKUSWB	PMAXSD	PSHUFW	REPNE	SETNZ	UNPCKLPS
BLENDPD	CMOVP	DIV	FIDIV	FSTENV	INVD	LES	MOVLPS	PADDB	PMAXSW	PSIGNB	REPNZ	SETO	VERR
BLENDPS	CMOVPE	DIVPD	FIDIVR	FSTP	INVEPT	LFENCE	MOVMSKPD	PADDD	PMAXUB	PSIGND	REPZ	SETP	VERW
BLENDVPD	CMOVPO	DIVPS	FILD	FSTP1	INVLPG	LFS	MOVMSKPS	PADDQ	PMAXUD	PSIGNW	RETF	SETPE	VMCALL
BLENDVPS	CMOVS	DIVSD	FIMUL	FSTPS	INVVPID	LGDT	MOVNTDQ	PADDSB	PMAXUW	PSLLD	RETN	SETPO	VMCLEAR
BOUND	CMOVZ	DIVSS	FINCSTP	FSTPS	IRET	LGS	MOVNTDQA	PADDSW	PMINSB	PSLLQ	ROL	SETS	VMLAUNCH
BSF	CMP	DPPD	FINIT	FSTSW	ISETD	LIDT	MOVNTPD	PADDUSB	PMINSD	PSLLQ	ROR	SETZ	VMPTRLD
BSR	CMPPD	DPPS	FIST	FSUB	JA	LLDT	MOVNTPS	PADDUSW	PMINSW	PSLLW	ROUNDPD	SFENCE	VMPTRST
BSWAP	CMPPS	DS	FISTP	FSUBP	JAE	LMSW	MOVNTPS	PADDW	PMINUB	PSRAD	ROUNDPS	SGDT	VMREAD
BT	CMPS	EMMS	FISTTP	FSUBR	JB	LOCK	MOVNTQ	PALIGNR	PMINUD	PSRAW	ROUNDSD	SHL	VMRESUME
BTC	CMPSB	ENTER	FISUB	FSUBRP	JBE	LODS	MOVQ	PAND	PMINUW	PSRLD	ROUNDSS	SHLD	VMWRITE
BTR	CMPSD	ES	FISUBR	FTST	JC	LODSB	MOVQ2DQ	PANDN	PMOVMSKB	PSRLDQ	RSM	SHR	VMXOFF
BTS	CMPSS	EXTRACTPS	FLD	FUCOM	JCXZ	LODSD	MOVS	PAUSE	PMOVSXBD	PSRLQ	RSQRTPS	SHRD	VMXON
CALL	CMPSW	F2XM1	FLD1	FUCOMI	JE	LODSW	MOVSB	PAVGB	PMOVSXBQ	PSRLW	RSQRTSS	SHUFPD	WAIT
CALLF	CMPXCHG	FABS	FLDCW	FUCOMIP	JECXZ	LOOP	MOVSD	PAVGW	PMOVSXBW	PSUBB	SAHF	SHUFPS	WBINVD
CBW	CMPXCHG8B	FADD	FLDENV	FUCOMP	JG	LOOPE	MOVSHDUP	PBLENDVB	PMOVSXDQ	PSUBD	SAL	SIDT	WRMSR
CDQ	COMISD	FADDP	FLDLZE	FUCOMPP	JGE	LOOPNE	MOVSLDUP	PBLENDW	PMOVSXWD	PSUBQ	SALC	SLDT	XADD
CLC	COMISS	FBLD	FLDL2T	FWAIT	JL	LOOPNZ	MOVSS	PCMPEQB	PMOVSXWQ	PSUBSB	SAR	SMSW	XCHG
CLD	CPUID	FBSTP	FLDL2G	FXAM	JLE	LOOPZ	MOVSW	PCMPEQD	PMOVZXBD	PSUBSW	SBB	SQRTPD	XGETBV
CLFLUSH	CRC32	FCHS	FLDLN2	FXCH	JMP	LSL	MOVSX	PCMPEQQ	PMOVZXBQ	PSUBUSB	SCAS	SQRTPS	XLAT
CLI	CS	FCLEX	FLDPI	FXCH4	JMPF	LSS	MOVUPD	PCMPEQW	PMOVZXBW	PSUBUSW	SCASB	SQRTSD	XLATB
CLTS	CVTDQ2PD	FCMOVB	FLDZ	FXCH7	JNA	LTR	MOVUPS	PCMPESTRI	PMOVZXDQ	PSUBW	SCASD	SQRTSS	XOR
CMC	CVTDQ2PS	FCMOVBE	FMUL	FXRSTOR	JNAE	MASKMOVDQU	MOVZX	PCMPESTRM	PMOVZXWD	PTEST	SCASW	SS	XORPD
CMOVA	CVTPD2DQ	FCMOVE	FMULP	FXSAVE	JNB	MASKMOVQ	MPSADBW	PCMPGTB	PMOVZXWQ	PUNPCKHBW	SETA	STC	XORPS
CMOVAE	CVTPD2PI	FCMOVNB	FNCLEX	FXTRACT	JNBE	MAXPD	MUL	PCMPGTD	PMULDQ	PUNPCKHDQ	SETAE	STD	XRSTOR
CMOVB	CVTPD2PS	FCMOVNBE	FNDISI	FYL2X	JNC	MAXPS	MULPD	PCMPGTQ	PMULHRSW	PUNPCKHQDQ	SETALC	STI	XSAVE
CMOVBE	CVTPI2PD	FCMOVNE	FNENI	FYL2XP1	JNE	MAXSD	MULPS	PCMPGTW	PMULHUW	PUNPCKHWD	SETB	STMXCSR	XSETBV
CMOVC	CVTPI2PS	FCMOVNU	FNINIT	GETSEC	JNG	MAXSS	MULSD	PCMPISTRI	PMULHW	PUNPCKLBW	SETBE	STOS	

Boolean axioms

Name	Definition
Identity law	1 & A = A 0 | A = A
Null law	0 & A = 0 1 | A = 1
Idempotent law	A & A = A A | A = A
Inverse law	A & !A = 0 A | !A = 1
Commutative law	A & B = B & A A | B = B | A
Associative law	(A&B) & C = A & (B&C) (A|B) | C = A | (B|C)
Distributive law	A | (B&C) = (A|B) & (A|C) A & (B|C) = (A&B) | (A&C)
Absorption law	A & (A|B) = A A | (A&B) = A
De Morgan's law	!(A&B) = !A | !B !(A|B) = !A & !B
Double negation law	!!A = A