

MIPS指令集格式 (1/6)

- 指令集是CPU所能執行指令的集合，指令集可分為複雜指令集和精簡指令集兩種。
 - 複雜指令集計算機(CISC)，ex：x86。
 - 精簡指令集計算機(RISC)，ex：MIPS。
- MIPS的所有指令長度皆相同(32bit)。
- 指令格式分成三種：
 - R-type：(register type)，與暫存器有關的指令格式。
 - I-type：(immediate type)，與資料傳送、分支、直接指令有關的格式。
 - J-type：(jump type)，與跳躍指令有關的格式。
- 以指令的第一個欄位(op)來判別指令格式。(參考最後兩頁)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R-type	op						rs				rt				rd				shamt				funct									
I-type	op						rs				rt				immediate																	
J-type	op						address																									

MIPS指令集格式 (2/6)

R-type指令 (1/2)

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
R-type	op						rs					rt					rd					shamt				funct						
I-type	op						rs					rt					immediate															
J-type	op						address																									

- op : (operation code)，指令運算碼。
- rs : (source operand register 1)，第一個來源運算元暫存器。
- rt : (source operand register 2)，第二個來源運算元暫存器。
- rd : (destination register)，目的運算元暫存器。
- shamt : (shift amount)，位移量(shift與rotate的次數)。
- funct : (function code)，功能碼(指定R-type功能，(參考最後1頁))。

MIPS指令集格式 (2/6)

R-type指令 (2/2)

\$17	\$s1	
\$18	\$s2	
\$19	\$s3	
\$31	\$ra	副程式返回位址

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R-type	op						rs						rt				rd				shamt				funct							
I-type	op						rs						rt				immediate															
J-type	op						address																									

指令	機械語言						組合語言					註解
	op	rs	rt	rd	shamt	funct	funct	rd	rs	rt	shamt	
加法	0	18	19	17	0	32	add	\$s1,	\$s2,	\$s3		\$s1 = \$s2 + \$s3
減法	0	18	19	17	0	34	sub	\$s1,	\$s2	\$s3		\$s1 = \$s2 - \$s3
邏輯AND	0	18	19	17	0	36	and	\$s1,	\$s2,	\$s3		\$s1 = \$s2 & \$s3
邏輯OR	0	18	19	17	0	37	or	\$s1,	\$s2,	\$s3		\$s1 = \$s2 \$s3
邏輯NOR	0	18	19	17	0	39	nor	\$s1,	\$s2,	\$s3		\$s1 = ~((\$s2 \$s3))
向左移位	0	0	18	17	10	0	sll	\$s1,	\$s2,		10	\$s1 = \$s2 << 10
向右移位	0	0	18	17	10	2	srl	\$s1,	\$s2,		10	\$s1 = \$s2 >> 10
小於就設定	0	18	19	17	0	42	slt	\$s1,	\$s2,	\$s3		\$s1 = 1 if (\$s2 < \$s3); \$s1 = 0 else
跳躍根據 \$ra暫存器	0	31	0	0	0	8	jr		\$ra			goto \$31儲存的位址

MIPS指令集格式 (3/6)

I-type指令 (1/2)

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
R-type	op						rs					rt					rd					shamt				funct						
I-type	op						rs					rt					immediate															
J-type	op						address																									

- op : (operation code) , 指令運算碼。
- rs : (base register) , 基址暫存器。
- rt : (target register) , 目標暫存器。
- immediate : 1) 常數(對於立即運算指令 , ex : addi)。
2) 記憶體相對位移量(對於記憶體讀寫指令 , ex : lw)。
3) pc相對定址位移量(對於跳躍分支指令 , ex : beq)。

MIPS指令集格式 (3/6)

I-type指令 (2/2)

\$17	\$s1
\$18	\$s2
\$19	\$s3

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
R-type	op						rs					rt					rd					shamt					funct						
I-type	op						rs					rt					immediate																
J-type	op						address																										

指令	機械語言				組合語言				註解
	op	rs	rt	immediate	op	rt	rs	immediate	
相等就跳躍	4	18	17	25	beq	\$s1,	\$s2,	100	if(\$s1==\$s2)goto pc+4+4*25
不相等就跳躍	5	18	17	25	bne	\$s1,	\$s2,	100	if(\$s2!=\$s2)goto pc+4+4*25
立即加法	8	18	17	100	addi	\$s1,	\$s2,	100	\$s1 = \$s2 + 100
立即小於就設定	10	18	17	100	slti	\$s1,	\$s2	100	\$s1=1 if (\$s2<100); \$s1 = 0 else
立即邏輯AND	12	18	17	100	andi	\$s1,	\$s2,	100	\$s1 = \$s2 & 100
立即邏輯OR	13	18	17	100	ori	\$s1,	\$s2,	100	\$s1 = \$s2 100
載入字組(word)	35	18	17	100	lw	\$s1,	100(\$s2)		\$s1= mem[\$s2 + 100]
儲存字組(word)	43	18	17	100	sw	\$s1	100(\$s2)		mem[\$s2 + 100] = \$s1

MIPS指令集格式 (4/6)

J-type指令

\$31	\$ra	副程式返回位址
------	------	---------

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
R-type	op						rs					rt					rd					shamt				funct						
I-type	op						rs					rt					immediate															
J-type	op						address																									

- op : (operation code) , 指令運算碼。
- address : 跳躍位址。

指令	機械語言		組合語言		註解
	op	address	op	address	
跳躍	2	2500	j	10000	goto 位址4*2500
跳躍和連結	3	2500	jal	10000	goto 位址4*2500且\$ra=pc+4

MIPS指令集格式 (5/6)

MIPS指令表 (1/2)

op	指令	註解	op	指令	註解
0	R-type (special operations 1)		16	cop0	coprocessor 0 operations
1	regimm		17	cop1	coprocessor 1 operations
2	j	jump	18	cop2	coprocessor 2 operations
3	jal	jump and link	19	cop1x	coprocessor 1 indexed operations
4	beq	branch equal	20	beql	branch equal likely
5	bne	branch not equal	21	bnel	branch not equal likely
6	blez	branch less equal zero	22	blezl	branch less equal likely
7	bgtz	branch great than zero	23	bgtzl	branch great than zero likely
8	addi	add immediate	24		
9	addiu	add immediate unsigned	25		
10	slti	set less than immediate	26		
11	sltiu	set less than immediate unsigned	27		
12	andi	and immediate	28	special operations 2	
13	ori	or immediate	29	jalx	jump and link exchange
14	xori	xor immediate	30		
15	lui	load upper immediate	31	special operations 3	

MIPS指令集格式 (5/6)

MIPS指令表 (2/2)

op	指令	註解	op	指令	註解
32	lb	load byte	48	ll	load linked word
33	lh	load half word	49	lwc1	load word to coprocessor 1
34	lwl	load word left	50	lwc2	load word to coprocessor 2
35	lw	load word	51	pref	prefetch
36	lbu	load byte unsighed	52		
37	lhu	load half unsigned	53	ldc1	load double word to coprocessor 1
38	lwr	load word right	54	ldc2	load double word to coprocessor 2
39			55		
40	sb	store byte	56	sc	store conditional word
41	sh	store half word	57	swc1	store word from coprocessor 1
42	swl	store word left	58	swc0	store word from coprocessor 0
43	sw	store word	59		
44			60		
45			61	sdc1	store double word from coprocessor 1
46	swr	store word right	62	sdc2	store double word from coprocessor 2
47	cache operations		63		

MIPS指令集格式 (6/6)

MIPS R-type指令表 (1/2)

funct	指令	註解	funct	指令	註解
0	sll	shift left logical	16	mfhi	move from HI register
1	movci	move conditional on floating point true	17	mthi	move to HI register
2	srl	shift right logical	18	mflo	move from LO register
3	sra	shift right arithmetic	19	mtlo	move to LO register
4	sllv	shift left logical variable	20		
5			21		
6	srlv	shift right logical variable	22		
7	srav	shift right arithmetic variable	23		
8	jr	jump register	24	mult	multiply
9	jalr	jump and link register	25	multu	multiply unsigned
10	movz	move conditional zero	26	div	divide
11	movn	move conditional not zero	27	divu	divide unsigned
12	syscall	system call	28		
13	break	break point	29		
14			30		
15	sync	synchronous	31		

MIPS指令集格式 (6/6)

MIPS R-type指令表 (2/2)

funct	指令	註解	funct	指令	註解
32	add	add	48	tge	trap if greater equal
33	addu	add unsigned	49	tgeu	trap if greater equal unsigned
34	sub	sub	50	slt	trap if less than
35	subu	sub unsigned	51	sltu	trap if less than unsigned
36	and	and	52	teq	trap equal
37	or	or	53		
38	xor	xor	54	tne	trap not equal
39	nor	nor	55		
40			56		
41			57		
42	slt	set less than	58		
43	sltu	set less than unsigned	59		
44			60		
45			61		
46			62		
47			63		

MIPS暫存器配置

暫存器號碼	暫存器名稱	用法		暫存器號碼	暫存器名稱	用法	
\$0	\$zero	常數值0		\$16	\$s0	儲存副程式變數值	保留
\$1	\$at	保留給組譯器使用		\$17	\$s1		
\$2	\$v0	儲存副程式回傳值	呼叫副程式時，不將暫存器儲存保留	\$18	\$s2		
\$3	\$v1			\$19	\$s3		
\$4	\$a0	儲存副程式參數值		\$20	\$s4		
\$5	\$a1			\$21	\$s5		
\$6	\$a2			\$22	\$s6		
\$7	\$a3			\$23	\$s7		
\$8	\$t0	儲存暫時值		\$24	\$t8	儲存暫時值	不保留
\$9	\$t1			\$25	\$t9		
\$10	\$t2			\$26	\$k0	保留給作業系統核心使用	
\$11	\$t3			\$27	\$k1		
\$12	\$t4			\$28	\$gp	全域指標	保留
\$13	\$t5			\$29	\$sp	堆疊指標	
\$14	\$t6			\$30	\$fp	程序框指標	
\$15	\$t7			\$31	\$ra	副程式返回位址	

Registers

Name	Register number	Usage	Preserved on call?
\$zero	0	The constant value 0	n.a.
\$v0-\$v1	2-3	Values for results and expression evaluation	no
\$a0-\$a3	4-7	Arguments	no
\$t0-\$t7	8-15	Temporaries	no
\$s0-\$s7	16-23	Saved	yes
\$t8-\$t9	24-25	More temporaries	no
\$gp	28	Global pointer	yes
\$sp	29	Stack pointer	yes
\$fp	30	Frame pointer	yes
\$ra	31	Return address	yes

Assembler Table (OP Code)

op(31:26)								
28-26	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
31-29								
0(000)	R-format	Bltz/gez	jump (j)	jump & link (jal)	branch eq (beq)	branch ne (bne)	blez	bgtz
1(001)	add immediate (addi)	addiu	set less than imm.	set less than imm. unsigned	andi	ori	xori	load upper immediate
2(010)	TLB	FlPt						
3(011)								
4(100)	load byte	load half	lwl	load word (lw)	load byte unsigned	load half unsigned	lwr	
5(101)	store byte	store half	swl	store word (sw)			swr	
6(110)	load linked word	lwcl						
7(111)	store cond. word	swcl						

R-Format (op=0b0000000)

op(31:26)=000000 (R-format), funct(5:0)								
2-0 5-3	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	shift left logical sll		shift right logical srl	sra	sllv		srlv	srav
1(001)	jump register jr	jalr			syscall	break		
2(010)	mfhi	mthi	mflo	mtlo				
3(011)	mult	multu	div	divu				
4(100)	add	addu	subtract	subu	and	or	xor	not or (nor)
5(101)			set l.t.	set l.t. unsigned				
6(110)								
7(111)								