

KGP_RISC

Register Usage Convention

Op Code	Op code representation	Function	Func Name	Function codes
Arithmetic	00	Add	add	0000
		Comp	comp	0001
		Multiply (unsigned)	multu	0010
		Multiply (signed)	mult	0011
		Add immediate	addi	0100
		Complement Immediate	compi	0101
Logic	00	AND	and	0110
		XOR	xor	0111
Shift	00	Shift left logical	shll	1010
		Shift right logical	shrl	1011
		Shift left logical variable	shllv	1110
		Shift right logical variable	shrlv	1111
		Shift right arithmetic	shra	1110
		Shift right arithmetic variable	shrav	1111
Memory	01	Store Word	sw	0000
	10	Load Word	lw	0000
Branch	11	Unconditional branch	b	0000
		Call	call	0001
		Branch Register	br	0011
		Branch on zero	bz	0100
		Branch on not zero	bnz	0101
		Branch on Sign	bs	0110
		Branch on Not Sign	bns	0111
		Branch on Carry	bcy	1001
		Branch on No Carry	bncy	1010
		Branch on Overflow	bv	1011
		Branch on No Overflow	bnv	1100
		Return	ret	0010

Instruction Format

Instruction Format				
OP	rs	rt	Func	Offset
2	5	5	4	16

Register Conventions

Register Number	Conventional Name	Usage	
\$0	\$zero	Hard-wired to 0	
\$1	\$ra	Return Address	
\$2 - \$3	\$v0, \$v1	Return values from functions	
\$4 - \$7	\$a0 - \$a3	Arguments to functions - not preserved by subprograms	
\$8 - \$15	\$t0 - \$t7	Temporary data, not preserved by subprograms	
\$16 - \$23	\$s0 - \$s7	Saved registers, preserved by subprograms	
\$24 - \$25	\$t8 - \$t9	More temporary registers, not preserved by subprograms	
\$26 - \$27	\$k0 - \$k1	Reserved for kernel. Do not use.	
\$28	\$gp	Global Area Pointer (base of global data segment)	
\$29	\$sp	Stack Pointer	
\$30	\$fp	Frame Pointer	
\$31	\$at	Reserved for pseudo-instructions	

Instruction Optimisation:

1. Arithmetic

Instruction	OP	rs	rt	Func	Offset
add, mult, multu,	00	5 bit	5 bit	4 bit	Don't care
addi, compi	00	5 bit	Don't care	4 bit	16 bit constant number

2. Logic

Instruction	OP	rs	rt	Func	Offset
and,xor	00	5 bit	5 bit	4 bit	Don't care

3. Shift

Instruction	OP	rs	rt	Func	Offset
shll, shrl, shllv, shrlv, shra, shrav	00	5 bit	5 bit Shift Amount	4 bit	Don't care

4. Memory

Instruction	OP	rs	rt	Func	Offset
lw	01	5 bit	5 bit	4 bit	16 bit
sw	10	5 bit	5 bit	4 bit	16 bit

5. Branch

Instruction	OP	rs	rt	Func	Offset
bz, bnz, bs, bns, br	11	5 bit	Don't care	4 bit	16 bit jump address
bcy, bncy, bv, bnv	11	5 bit	5 bit	4 bit	16 bit jump address
b, call	11	Don't care	Don't care	4 bit	16 bit jump address
ret	11	Don't care	Don't care	4 bit	Don't care