

KGP-RISC

Siddhant Agarwal (17CS30035), Divyang Mittal (17CS10012)

October 2019

1 Instruction Set Architecture (ISA)

Class	Instruction	Usage	Meaning
Arithmetic	Add	add rs,rt	$rs \leftarrow (rs) + (rt)$
	Multiply (unsigned)	multu rs,rt	$\{reg_{19}, reg_{20}\} \leftarrow (rs) \times_{unsigned} (rt)$
	Multiply (signed)	mult rs,rt	$\{reg_{19}, reg_{20}\} \leftarrow (rs) \times_{signed} (rt)$
	Comp	comp rs,rt	$rs \leftarrow 2's \text{ Complement } (rt)$
	Add immediate	addi rs,imm	$rs \leftarrow (rs) + imm$
	Complement Immediate	compi rs,imm	$rs \leftarrow 2's \text{ Complement } (imm)$
Logic	AND	and rs,rt	$rs \leftarrow (rs) \wedge (rt)$
	XOR	xor rs,rt	$rs \leftarrow (rs) \oplus (rt)$
Shift	Shift left logical	shll rs, sh	$rs \leftarrow (rs)$ left-shifted by sh
	Shift right logical	shrl rs, sh	$rs \leftarrow (rs)$ right-shifted by sh
	Shift left logical variable	shllv rs, rt	$rs \leftarrow (rs)$ left-shifted by (rt)
	Shift right logical variable	shrlv rs, rt	$rs \leftarrow (rs)$ right-shifted by (rt)
	Shift right arithmetic	shra rs, sh	$rs \leftarrow (rs)$ arithmetic right-shifted by sh
	Shift right arithmetic variable	shrav rs, rt	$rs \leftarrow (rs)$ right-shifted by (rt)
Memory	Load Word	lw rt,imm(rs)	$rt \leftarrow mem[(rs) + imm]$
	Store Word	sw rt,imm,(rs)	$mem[(rs) + imm] \leftarrow (rt)$
Branch	Unconditional branch	b L	goto L
	Branch Register	br rs	goto (rs)
	Branch on zero	bz L	if ($zflag == 1$) then goto L
	Branch on not zero	bnz L	if ($zflag == 0$) then goto L
	Branch on Carry	bcy L	if ($carryflag == 1$) then goto L
	Branch on No Carry	bncy L	if ($carryflag == 0$) then goto L
	Branch on Sign	bs	if ($signflag == 1$) then goto L
	Branch on Not Sign	bns L	if ($signflag == 0$) then goto L
	Branch on Overflow	bv L	if ($overflowflag == 1$) then goto L
	Branch on No Overflow	bnv L	if ($overflowflag == 0$) then goto L
	Call	Call L	$ra \leftarrow (PC)+4$; goto L
	Return	Ret	goto (ra)

2 Register Usage Convention

Register	Function	Register Number	Register Code
\$zero	0 register, stores the constant 0	0	00000
\$t0 - \$t11	Caller saved, not preserved across function calls	1 - 12	00001 - 01100
\$v0 - \$v1	Return values from functions	13 - 14	01101 - 01110
\$a0 - \$a3	Parameters to function call	15 - 18	01111 - 10010
\$hi	Most significant word of multiplication	19	10011
\$lo	Least significant word of multiplication	20	10100
\$s0 - \$s8	Saved variables, preserved during function calls	21 - 29	10101 - 11101
\$sp	Stack Pointer	30	11110
\$ra	Register to store return address	31	11111

3 Instruction Format and Encoding

The various instructions in the KGP-RISC ISA can be categorised into the following four categories (called R-Format, I-Format, J1-Type, J2-Type)

Opcode[1:0]	Binary Representation	Format	Functions
0	00	R-Format	add, multu, mult, comp, and, xor, shll, shrl, shllv, shrlv, shra, shrav
1	01	I-Format	compi, addi, lw, sw
2	10	J1-Type	br, ret
3	11	J2-Type	b, bz, bnz, bcy, bncy, bs, bns, bv, bnv, Call

3.1 Opcode 00 00XX: R-Format Instructions

Opcode (6 bits)	rs (5 bits)	rt (5 bits)	shamt (5 bits)	Function (3 bits)	Dont Care (8 bits)
--------------------	----------------	----------------	-------------------	----------------------	-----------------------

Function Codes

Function	Function Codes	Opcode
add	000	00 0000
mult	001	00 0000
multu	010	00 0000
comp	000	00 0001
and	001	00 0001
xor	010	00 0001
shll	000	00 0010
shrl	001	00 0010
shllv	010	00 0010
shrlv	011	00 0010
shra	100	00 0010
shrav	101	00 0010

3.2 Opcode 01 00XX: I-Format Instructions

Opcode (6 bits)	rs (5 bits)	rt (5 bits)	Immediate (16 bits)
--------------------	----------------	----------------	------------------------

Function Codes

Function	Opcode
compi	01 0000
addi	01 0001
lw	01 0010
sw	01 0011

3.3 Opcode 10 000X: J1-Type Instructions

Opcode (6 bits)	reg (5 bits)	Don't Care (21 bits)
--------------------	-----------------	-------------------------

Function Codes

Function	Opcode
br	10 0000
Ret	10 0001

3.4 Opcode 11 0XXX: J2-Format Instructions

Opcode (6 bits)	L (26 bits)
---------------------------	-----------------------

Function Codes

Function	OpCode
b	11 0000
bz	11 0001
bnz	11 0010
bcy	11 0011
bncy	11 0100
bs	11 0101
bns	11 0110
bv	11 0111
bnv	11 1000
Call	11 1001