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 Complement (rt)
	Add immediate	addi rs,imm	$rs \leftarrow (rs) + imm$
	Complement Immediate	compi rs,imm	$rs \leftarrow$ 2's 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
$v0 - $v1	Saved variable, return values from functions	1 - 2	00001 - 00010
$a0 - $a3	Parameters for a function call	3 - 6	00011 - 00110
$t0 - $t11	Temporaries	7 - 18	00111 - 10010
$lo	Most significant word of multiplication	19	10011
$hi	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 six categories (called R-Format, I-Format, Memory-Access-Format, J1-Type, J2-Type, J3-Type)

Opcode	Binary Representation	Format	Functions
0	000	R-Format	add, multu, mult, comp, and, xor, shll, shrl, shllv, shrlv, shra, shrav
1	001	I-Format	compi, addi
2	010	Memory Acess	lw, sw
3	011	J1-Format	bz, bnz, bcy, bncy, bs, bns, bv, bnv
4	100	J2-Format	b, Call
5	101	J3-Format	Ret, br

3.1 Opcode 000: R-Format Instructions

Opcode (3 bits)	rs (5 bits)	rt (5 bits)	shamt (5 bits)	Function (4 bits)	Dont Care (10 bits)

Function Codes

Function	Function Codes	Binary Representation
add	0	0000
mult	1	0001
multu	2	0010
comp	3	0011
and	4	0100
xor	5	0101
shll	6	0110
shrl	7	0111
shllb	8	1000
shrlb	9	1001
shra	10	1010
shrab	11	1011

3.2 Opcode 001: I-Format Instructions

Opcode (3 bits)	**rs** (5 bits)	**Immediate** (22 bits)	**Function** (2 bits)

Function Codes

Function	Function Code	Binary Representation
compi	0	00
addi	1	01

3.3 Opcode 010: Memory Access Instructions

Opcode (3 bits)	**rs** (5 bits)	**rt** (5 bits)	**Immediate** (18 bits)	**Function** (1 bits)

Function Codes

Function	Function Code	Binary Representation
lw	0	0
sw	1	1

3.4 Opcode 011: J1-Format Instructions

Opcode (3 bits)	**Function** (3 bits)	**L** (26 bits)

Function Codes

Function	Function Code	Binary Representation
bz	0	0000
bnz	1	0001
bcy	2	0010
bncy	3	0011
bs	4	0100
bns	5	0101
bv	6	0110
bnv	7	0111

3.5 Opcode 100: J2-Format Instructions

Opcode (3 bits)	**Function** (1 bits)	**L** (26 bits)	**Don't Care** (2 bits)

Function Codes

Function	Function Code	Binary Representation
Call	0	0
b	1	1

3.6 Opcode 101: J3-Format Instructions

Opcode (3 bits)	**Function** (1 bits)	**reg** (5 bits)	**Don't Care** (23 bits)

Function Codes

Function	Function Code	Binary Representation
Ret	0	0
br	1	1