KGP RISC ISA (Group - 26)

- Binary Instruction - 32 bit
- Opcode - 4 bits
- Function Code - 4 bits (except for Load Word and Store Word which has no Function Code)
- Every register has a 5-bit code.

REGISTERS:

R0 -	zero	- always contains 0
R1 -	at	- Assembler Temporary
R2-R3	v0 - v1	- Function return value
R4-R7	a0-a3	- Function Parameters
R8-R14	t0-t6	- Function temporary values
R15	gp	- Global Pointer
R16	sp	- Stack Pointer
R17	Flag_reg	- R17[0] - zflag R17[1] - carryflag R17[2] - signflag R17[3] - Overflowflag
R18	S8	- Saved register across function calls
R19	Hi	- used to hold the most significant word in mult and div results
R20	Lo	- used to hold the least significant word in mult and div results
R21-R26	S0-S5	- saved registers across function calls
R27-R28	K0-K1	- reserved for interrupt handler
R29	ra	- return address from function call
R30	PC	- Program Counter, points at 8 bytes past current instruction
R31	EPC	- Exception Program Counter

INSTRUCTION SET BIT FORMAT:

- Instructions in each table below have unique op - code

Arithmetic: (OpCode - 0000)

INSTRUCTION	INSTRUCTION (BINARY FORMAT)
Add	0000 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0000 (OpCode) (shift) (zeros) (Fn Code)
Multiply (Unsigned)	0000 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0001
Multiply (signed)	0000 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0010
Complement	0000 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0011

Add immediate	0000 (rs - 5-bits) (immediate - 16-bits) 000 0100
Complement immediate	0000 (rs - 5-bits) (immediate - 16-bits) 000 0101

Logic : (Opcode - 0001)

INSTRUCTION	**INSTRUCTION (BINARY FORMAT)**
AND	0001 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0000 (OpCode) (shift) (8 zeros) (Fn Code)
XOR	0001 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0001

Shift-1 (shift given in binary - sh) : (Opcode - 0010)

INSTRUCTION	**INSTRUCTION (BINARY FORMAT)**
Shift Left Logical	0010 (rs - 5-bits) 00000 (sh - 6-bit shift) 00000000 0000
Shift right logical	0010 (rs - 5-bits) 00000 (sh - 6-bit shift) 00000000 0001
Shift right arithmetic	0010 (rs - 5-bits) 00000 (sh - 6-bit shift) 00000000 0010

Shift - 2 (shift variable) : (Opcode - 0011)

INSTRUCTION	**INSTRUCTION (BINARY FORMAT)**
Shift Left Logical variable	0011 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0000
Shift right logical variable	0011 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0001
Shift right arithmetic variable	0011 (rs - 5-bits) (rt - 5-bits) 000000 00000000 0010

Load Word : (Opcode - 0100)

INSTRUCTION	**INSTRUCTION (BINARY FORMAT)**
Load Word	0100 (rt - 5-bits) (rs - 5-bits) (Immediate - 16 bits) 00

Store Word : (Opcode - 0101)

INSTRUCTION	INSTRUCTION (BINARY FORMAT)
Store Word	0101 (rt - 5-bits) (rs - 5-bits) (Immediate - 16 bits) 00

Branch-1: (Opcode - 0110)

INSTRUCTION	INSTRUCTION (BINARY FORMAT)
Unconditional Branch	0110 (Offset - 16 bits) 00000000(zeros) 0000(Fn-code)
Branch Register	0110 (rs - 5-bits) (19 - zeros) 0001

Branch - 2: (Opcode - 0111)

INSTRUCTION	INSTRUCTION (BINARY FORMAT)
Branch on Zero	0111 (1 - bit -zflag) (16 - bit Offset) 0000000 0000(Fn-Code)
Branch on Not Zero	0111(1 - bit -zflag) (16 - bit Offset) 0000000 0001
Branch on Carry	0111(1- bit -carryflag) (16 - bit Offset) 0000000 0010
Branch on No Carry	0111(1- bit -carryflag) (16 - bit Offset) 0000000 0011
Branch on Sign	0111(1- bit -signflag) (16 - bit Offset) 0000000 0100
Branch on Not Sign	0111(1- bit -signflag) (16 - bit Offset) 0000000 0101
Branch on Overflow	0111(1 - bit -Overflowflag) (16 - bit Offset) 0000000 0110
Branch on No Overflow	0111(1 - bit -Overflowflag) (16 - bit Offset) 0000000 0111

Branch - 3: (Opcode - 1000)

INSTRUCTION	INSTRUCTION(BINARY FORMAT)
Call	1000 (16 - bit Offset) 00000000 0000 (Function Code)
Return	1000 (24 - zeros) 0001