指令							format	functional
J		b'000010	instr_index				J target	PC + offset
JR		b'000000	rs	b'00000	b'00000	hint	b'001000	JR rs	PC <- rs
BEQ		b'000100	rs	rt	offset			BEQ rs, rt, offset	if rs = rt then PC + offset
BLTZ		b'000001	rs	b'00000	offset			BLTZ rs, offset	if rs < 0 then PC + offset
BGTZ		b'000111	rs	b'00000	offset			BGTZ rs, offset	if rs > 0 then PC + offset
BREAK		b'000000	code				b'001101	BREAK	BREAK
SW		b'101011	base	rt	offset			SW rt, offset(base)	mem[base + offset] <- rt
LW		b'100011	base	rt	offset			LW rt, offset(base)	rt <- mem[base + offset]
SLL		b'000000	b'00000	rt	rd	sa	b'000000	SLL rd, rt, sa	rd <- rt << sa
SRL		b'000000	b'00000	rt	rd	sa	b'000010	SRL rd, rt, sa	(logical) rd <- rt >> sa
SRA		b'000000	b'00000	rt	rd	sa	b'000011	SRA rd, rt, sa	(arithmetic) rd <- rt >> sa
NOP		b'000000	b'00000	b'00000	b'00000	b'00000	b'000000	NOP	NOP
ADD	<- rs + st	b'000000	rs	rt	rd	b'00000	b'100000	ADD rd, rs, rt	rd <- rs + st
SUB	<- rs - st	b'000000	rs	rt	rd	b'00000	b'100010	SUB rd, rs, rt	rd <- rs - st
MUL		b'011100	rs	rt	rd	b'00000	b'000010	MUL rd, rs, rt	rd <- rs * rt
AND		b'000000	rs	rt	rd	b'00000	b'100100	AND rd, rs, rt	rd <- rs AND rt
NOR		b'000000	rs	rt	rd	b'00000	b'100111	NOR rd, rs, rt	rd <- rs NOR rt
SLT	(rs < rt)	b'000000	rs	rt	rd	b'00001	b'101010	SLT rd, rs, rt	rd <- (rs < rt)