Registers

Number	Name	Description
0	X0	Always is equal to 0
1	X1	General purpose register
2	X2	General purpose register
3	X3	General purpose register
4	X4	General purpose register
5	X5	General purpose register
6	X6	Stack pointer
7	X7	In this register is store the return address when a jal instruction is executed

Instructions

Nemonic	Opcode	Operation	Description
add	0000	R[rd] = R[rs] + R[rt]	Register rs is added to register rt and result is store in register rd.
addi	0001	R[rt]= R[rs] + Imm	Register rs is added to Imm and result is store in register rd. Where Imm is constant value in two's complement.
and	0010	R[rd]=R[rs] & R[rt]	It´s a and bitwise between Register rs and rt and result is store in register rd.
andi	0011	R[rt]=R[rs] & Imm	It´s a and bitwise between Register rs and Imm and result is store in register rd. Where Imm is a constant value in two's complement.
beq	0100	if(R[rt]==R[rs]) PC = PC + BranAddr	If rt and rs are equal the PC = PC + BranAddr. Where BranAddr is the branch destination (offset) in two's complement.
bne	0101	if(R[rd]!=R[rs1]) PC = PC + BranAddr	If rt and rs are not equal the PC = PC + BranAddr. Where BranAddr is the branch destination (offset) in two's complement.
j	0110	PC = Addr	Unconditional jump to Addr
jal	111	PC = JumAddr; X7= PC+1	It's is for procedures, when it´s executed PC gets the jump address and X7 stored current PC +1.
jr	1010	PC= X7	It's used to return from procedures; when it´s executed PC gets the value stored in X7.

lb	1011	R[rt] = M[rs] + Imm]	It’s a load byte from memory. The memory addres is calculate by adding rs and Imm; where Imm is a constant value in two’s complement and rt is the destination register.
or	1100	R[rd]=R[rs] | R[rt]	It´s a or bitwise between Register rs and rt and result is store in register rd.
sb	1101	M[rs + Imm] = R[rt]	It’s a stored byte from memory. The memory addres is calculate by adding rs and Imm; where Imm is a constant value in two’s complement and rt is the source register.
sll	1110	R[rd] = R[rt] << R[rs]	Shift logical left; rt is shifted to the left base on rs
srl	1111	R[rd] = R[rt] >> R[rs]	Shift logical right; rt is shifted to the rigth base on rs