Edited VERSioN

ISA design 2.0

Highlighted the changed parts

PC	Functionality	Instruction	Coding	Example	Ex ML
PC++	Rx = Mem[Ry]	Lw rx,ry	000 xx yy	Lw R1,R1	0000101
PC++	Mem[Ry] = Rx	Sw Rx, Ry	001 xx yy	Sw R1,R1	0010101
PC++	Rx+Ry	Add Rx,Ry	010 xx yy	Add R1,R2	0100110
PC++	If Rx < Ry then R0 = 1 Otherwise R0=0	Slt Rx,Ry	0110 x yy	Slt R1,R2	0110010
PC= PC+imm	If R(0,1)=0 then PC + R(0,1,2,3)	Beqz Rx, Ry	0111 x yy	beqz R1,R3	0111111
PC++	Rx-Ry	Sub Rx,Ry	100 xx,yy	Sub R1,R1	1000101
PC++	Rx = Ry AND 1	Andi Rx,Ry	101 xx yy	Andi R0,R2	1010010
PC++	Rx = imm [-4,3]	Li Rx iii	110 x iii	Li R0, -4	1100111
PC++	Rx = Rx-1	Subi Rx	11100 Rx	Subi R3	1110011
PC++	R0=R0 XOR Rx	XOR Rx	11101 Rx	Xor R1	1110101
PC++	Rx >> 1	SRL Rx	11110 Rx	Srl R2	1111010
PC= PC+Rx	PC+Rx	J Rx	11111 Rx	J R2	1111110
PC = 0	End Program	HALT	0111000	N/A	N/A

1. Justification for Beqz Change:
    a. Provides beqz with more flexibility with what it's comparing and give it the ability to jump anywhere
    b. Provides more options

Will have to use Beqz less frequently thus reducing the DIC

ISA 1.0 OLD

PC	Functionality	Instruction	Coding	Example	Ex ML
PC++	Rx = Mem[Ry]	Lw rx,ry	000 xx yy	Lw R1,R1	0000101
PC++	Mem[Ry] = Rx	Sw Rx, Ry	001 xx yy	Sw R1,R1	0010101
PC++	Rx+Ry	Add Rx,Ry	010 xx yy	Add R1,R2	0100110
PC++	If Rx < Ry then R0 = 1 Otherwise R0=0	Slt Rx,Ry	0110 x yy	Slt R1,R2	0110010
PC= PC+imm	If R0=0 then PC + imm [0,7]	Beqz iii	0111 iii	beqz 2	0111010
PC++	Rx-Ry	Sub Rx,Ry	100 xx,yy	Sub R1,R1	1000101
PC++	Rx = Ry AND 1	Andi Rx,Ry	101 xx yy	Andi R0,R2	1010010
PC++	Rx = imm [-4,3]	Li Rx iii	110 x iii	Li R0, -4	1100111
PC++	Rx = Rx-1	Subi Rx	11100 Rx	Subi R3	1110011
PC++	R0=R0 XOR Rx	XOR Rx	11101 Rx	Xor R1	1110101
PC++	Rx >> 1	SRL Rx	11110 Rx	Srl R2	1111010
PC= PC+Rx	PC+Rx	J Rx	11111 Rx	J R2	1111110
PC = 0	End Program	HALT	0111000	N/A	N/A