Final

Submitted by: Piyusha Jahagirdar

AND COMPLEMENT IMMEDIATE

`andci rt, rs,` ***Immediate***

The ANDCI instruction takes operand A form the RS register, operand B form the extended 16-bits and will store the result in the RT register. As we will not need the RD register here, the *RegDest* will be 0. As we will have to load the data into the Registers after the clock signal the signal *RegWrite* will be 1. The data stored in the RS register (A operand) will be available at the RD1 register which will be an input to the ALU, the other operand (B operand) will be from the (sign-extended) lower 16 bits of the instruction. Therefore, the *ALUSrc* will 1. The *ALUControl* will be set to 100 as it is the code for the ANDCI instruction. The result from the ALU will be stored to the RT register, so the *MemWrite* and *MemtoReg* both will be 0. As there is no branching and looping, the *Branch* and *Jump* both will also be 0.

Values of Control Signal

Signal	**Value**
MemtoReg	0
MemWrite	0
Branch	0
ALUControl	100
ALUSrc	1
RegDst	0
RegWrite	1
Jump	0