Comp Arch HW #2

4.2.1: The Existing blocks that can be used are:

(a) This instruction, uses instruction memory, both existing read ports of registers, the ALU (to be able to compare Rs and Rt) and the write port of Registers.

(b) This instruction, also uses instruction memory, both registers read ports, ALU to add Rd and Rs together, uses data memory and uses the write ports in Registers.

4.2.2: The new functional blocks needed are:

(a) This instruction needs the 0 output of the ALU to be zero extended in order for it to compute the value for Rd. Afterwards we need to put this as another input to the Mux that selects which value should be written into the Registers.

(b) The instruction doesn't need any, The instruction can simply be implemented using the existing blocks.

4.2.3: The new control signals are:

(a) Here you need a new control signal for the Mux that picks between values that can be written into Registers.

(b) Here you don't need any new control signal. This instruction can be implemented without it, it only requires changes in the control Magic.

4.1.1: The values of the signals

	RegWrite	MemRead	ALUMux	MemWrite	ALUop	Reg Mux	Branch
(a)	1	0	0 (Reg)	0	AND	1 (ALU)	0
(b)	0	0	1 (Imm)	1	ADD	x	0

ALUMux ~~Sel~~ – the control signal that controls Mux at the ALU input, 0 (Reg) chooses the output of the register file and 1 (Imm) selects the immediate from the instruction word as the second input to the ALU.

Reg Mux – the control signal that controls the Mux at the data input to the register, 0 (ALU) selects the output of the ALU, and 1 (Mem) selects the output of memory.

A value of x is a 'don't care' (does not matter if signal is 0 or 1)

4.1.2: Resources performing a useful function for this instruction:

(a) All of them except Data Memory and Branch Add unit.

(b) All of them except branch Add unit and write port of the Registers.

4.1.3:

	outputs that aren't used	No outputs
(a)	branch Add	Data Memory
(b)	branch add, write port of Registers	None (all units produce outputs)