

## Implementing MIPS Single Cycle Control

5/10/2002

129

## Introduction

- These instructions are of interest:
  - Arithmetic/Logical: ADD, SUB, MULT, DIV, AND, OR, XOR
  - Arithmetic/Logical I-type: ADDI, ORI
  - Control: BEQ, BNE, J, JAL, JR
  - Memory: LW, SW, LB, SB

5/10/2002

130

## Main Control Unit

- Has to determine the following control signals:
  - RegDest: which destination register to write
  - Data2Reg: which data to write back to the register file?
  - RegWrite: should the register file be written?
  - ALUSrc: which source operand for the ALU?
  - MemRead/MemWrite/MemByte: memory control...
  - Control for branches and jumps: (homework)
  - Possibly an ALU operation: (homework)

5/10/2002

131

## Truth Table for the Main Control

- We can write the truth table for the Main control unit.
- It is a function of the OPCODE field and the FUNC field.
- We can implement it using a ROM, PLA, etc...

5/10/2002

132

## Controlling the ALU

- Sometimes the main control unit knows what operation the ALU should do.
  - For which instructions can it look at just the OPCODE field and tell what the ALU should be doing?
  - For which instructions does it need to look at the FUNC field?

5/10/2002

133

## ALU Control Issues

- The book uses the following strategy:
- The main control sends a signal to the ALU control unit, which means:
  - This is the op to perform (ADD, SUB, OR...)
  - otherwise, you decide the operation...
- Since we've already got both the FUNC and OP field coming into the main control unit, we could, in principle, decide everything there...

5/10/2002

134