Control Unit (single cycle implementation)

• Control unit sends control signals to data path and memory depending
  – on the opcode (and function field)
  – results in the ALU (for example for Zero test)
• These signals control
  – muxes; read/write enable for registers and memory etc.
• Some “control” comes directly from instruction
  – register names
• Some actions are performed at every instruction so no need for control (in this single cycle implementation)
  – incrementing PC by 4; reading instr. memory for fetching next inst.
Building the control unit

• Decompose the problem into
  – Data path control (register transfers)
  – ALU control

• Setting of control lines by control unit totally specified in the ISA
  – for ALU by opcode + function bits if R-R format
  – for register names by instruction
  – for reading/writing memory and writing register by opcode
  – muxes by opcode
  – PC by opcode + result of ALU
Implementation

- Input: opcode
- Output: setting of control lines
- Can be done by logic equations
- If not too many, like in RISC machines
  - Use of PAL’s (cf. CSE 370).
  - In RISC machines the control is “hardwired”
- If too large (too many states etc.)
  - Use of microprogramming (a microprogram is a hardwired program that interprets the ISA)
- Or use a combination of both techniques (Pentium)
Where are control signals needed (cf. Figure 5.17)

• Register file
  – RegWrite (Register write signal for R-type, Load)
  – RegDst (Register destination signal: rd for R-type, rt for Load)

• ALU
  – ALUSrc (What kind of second operand: register or immediate)
  – ALUop (What kind of function: ALU control for R-type)

• Data memory
  – MemRead (Load) or MemWrite (Store)
  – MemtoReg (Result register written from ALU or memory)

• Branch control
  – PCSrc (PC modification if branch is taken)
How are the control signals asserted (cf. Fig 5.19)

• Decoding of the opcode by control unit yields
  – Control of the 3 muxes ($\text{RegD}st$, $\text{ALUS}rc$, $\text{Mem}to\text{Reg}$): 3 control lines
  – Signals for $\text{RegWrite}$, $\text{Memread}$, $\text{Memwrite}$: 3 control lines
  – Signals to activate ALU control (e.g., restrict ourselves to 2)
  – Signal for branch (1 control line)
    • decoding of opcode ANDed with ALU zero result

• Input Opcode: 6 bits

• Output 9 control lines (see Figure 5.27)