# Control Unit for single cycle implementation

Function:

- Decodes instruction opcode/function field
- Sends signals to the data path (muxes) and to the ALU

Some "controls" come directly from the instruction

- Register fields indicate which registers to read/write
- Immediate field comes directly from the instruction also

Some actions (e.g., incrementing the PC by 4, reading the instruction memory to fetch the next instruction) are performed for every instruction and do not need to be explicitly represented in this first pass at implementation.

# Implementing the Control Unit

Building the control is not that complicated.

Decompose the problem into ALU control and data path control.

In both cases

- Input signals are specified in the ISA
  - for ALU control they are: some bits of the opcode plus, if needed, function bits in R-R format.
  - for Control Unit: opcode
- Output signals can be defined as a function of the input signals
- Use of PLA's (or PAL's. cf. CSE 370) to build the "hardwired" control This is what is used in RISC machines.
- Another option, for complex instructions, is "Microprogramming" (see later)
- Some modern machines like Pentium Pro use a combination of hardwired control for "simple" instructions and microprogramming for complex instructions (like instructions manipulating strings)

# Where are control signals needed (cf. Figure 5.17 in your book)

1. Register file
   (a) Register write signal (R-R operations, Load instructions etc.)
   (b) Which register destination field to use (rt or rd; different for a Load and an R-instruction)
2. ALU
   (a) What kind of second operand (register or immediate) is needed
   (b) What kind of function (add, subtract, AND etc.) will be performed
3. Memory
   (a) Read signal
   (b) Write signal
   (c) What is written in the result register (different for an R-instruction and for a Load)
4. Branch control
   (a) Modifying the PC in case of a successful branch

# How are the control signals asserted (cf. Figure 5.19 of your book)

Control unit gets opcode. Decoding yields:

1. Controlling the 3 MUXes (Register destination, ALU source, source of register write)
2. Signals for read/write memory and register write
3. Signal to active ALU control (this involves bits of opcode and the function bits)
4. Signal for branch. This has to be ANDed with output of ALU to either use incremented PC or new PC.