Processor Control

Adding Control

- Control Unit:
  - Decodes instruciton opcode/function field
  - Sends signals to the data path (muxes, reg file, memories)
- Some controls come directly from the instruction:
  - Register fields indicate which register to read/write
  - Immediate field
- Building the control unit is not that complicated:
  - Input signals (opcode/function) are specified by the ISA
  - Output signals can be identified easily from the opcode
  - We can use PLAs (see CSE370) to build hardwired control units

Review of Instruction Format

- The opcode lives in bits 31-26
- The two registers to read are always the rs (25-21) and rt (20-16) registers
- For a load/store, we find the base register in the rs field (25-21)
- The 16 bit offset (for branch or load/store) is in 15-0
- The destination register can be in one of two places:
  - Loads: rt field (20-16)
  - R-type: rd field (15-11)
  - This implies we'll need a mux to select between these two fields.

Where are control signals needed?

- Register File:
  - *RegDst* - selects between rt and rd field as destination register (different for Load-store than R-type)
  - *RegWrite* - do we want to write a register? (R-type, Load-store)
- ALU:
  - *ALUSrc* - selects between immediate or register value as source for ALU (different for R-type and I-type)
  - Also need to select kind of ALU operation (bits 0-5 is function)
- Memory:
  - *MemWrite* - are we writing memory? (store instructions)
  - *MemRead* - are we reading memory? (load instructions)
  - *MemToReg* - selects between memory value or ALU output as writeback to the register file
- *Branch* - are we branching?

How are signals asserted?

- Control unit gets the opcode. Decoding yields:
  - Control for the 3 muxes (RegDst, ALUSrc, MemToReg)
  - Signals for read/write memory
  - Signal for register write
  - Signal for branch (ANDed with output of ALU)
  - Signal to ALU Control unit
- We also have a small control unit for the ALU, which takes the signal from the main control unit, together with the funct field (5-0) from the instruction.
- We'll focus on the main control unit. The ALU control is similar.

Examples

- Using the provided figure (5.22), fill in this table which specifies the control signals for the various instructions:

Instr.	opcode bits: 543210	Reg Dst	ALU Src	Mem to Reg	Reg Write	Mem Read	Mem Write	Br
R-format	000000							
lw	100011							
sw	101011							
beq	000100							

Control Functions

- We can express the functions for the control lines logically:
  - RegDst = !op5 AND !op2
  - ALUSrc = op5
  - MemtoReg = op5
  - RegWrite = !op2 AND !op3
  - MemRead = op5 AND !op3
  - MemWrite = op5 AND op3
  - Branch = op2
- For our subset instruction set, this minimized logic is fine, but for the full instruction set (64 opcodes), we'd want something more general: We would specify the truth table, and implement it using a PLA.