Multicycle datapath implementation

- Once clock cycle per execution step
- Instructions will take a variable number of clock cycles (3-5 for MIPS)
- Functional units are now reused in the architecture
- Registers added to store output of previous cycle for next cycle(s)

Additional registers

- IR: Instruction register
  - needs explicit write signal
- MDR: Memory data register
- A, B: operands read from register file
- ALUOut: output of ALU

Note: Except for IR, all registers are written on each clock cycle.
Execution steps: All instructions

- Step 1: Fetch instruction, store in IR, increment PC
- Step 2:
  - Decode instruction
  - “Look ahead” steps:
    - Read in rs and rt registers to A and B, respectively
    - Compute target branch address using lower 16 bits of instruction --> ALUOut

Execution steps: beq

- Step 3:
  - Subtract two registers
  - Use Zero output to determine if they are equal
  - Determine if we branch to the address in ALUOut or to PC+4

Execution steps: R-type

- Step 3: Perform specified operation --> ALUOut
- Step 4: Write ALUOut contents to specified register

Execution steps: Memory access

- Step 3: Compute memory address
- Step 4:
  - If lw: Retrieve data from memory at specified address and place in MDR
  - If sw: Write data (B register) to memory at specified address
- Step 5 (lw only): Write contents of MDR to specified register
Multicycle control unit

Finite state machine (Moore)

Control unit settings

- **ALUSrcA**
  - 0 = First ALU operand is PC
  - 1 = First ALU operand is A
- **MemtoReg**
  - 0 = Data to write to register is from ALUOut
  - 1 = Data to write to register is from MDR
- **IorD**
  - 0 = PC supplies address to memory unit
  - 1 = ALUOut supplies address to memory unit
- **PCWrite**
  - PC written, source controlled by PCSource
- **PCWriteCond**
  - PC written if Zero output is true

ALU control signals

- **ALUOp**
  - 00: Add
  - 01: Subtract
  - 10: Read function field
- **ALUSrcB**
  - 00: Second input is from B
  - 01: Second input is 4
  - 10: Second input is sign-extended lower 16 bits of IR
  - 11: Same as 10, shifted left by 2
- **PCSource**
  - 00: PC = PC+4
  - 01: PC = branch target address
  - 10: PC = jump target address

IA-32 datapath and control

- Multicycle datapath with *superscalar* pipelining
- Microprogrammed control
  - hardwired for simple instructions
  - microcoded for complex instructions
- Note: IA-32's instructions may take 10's or 100's of clock cycles to execute!
Microinstructions

- Smaller “sub-instructions” of more complex instructions
- Up to 3 microinstructions are stored in the trace cache (buffer)
- The rest of the microinstructions are stored in a specialized ROM
- Decoder accesses the correct sequence of microinstructions

More about the IA-32 implementation

- Some datapath resources are duplicated
  - “multiple datapaths” in the processor (one per instruction class)
- Sometimes a simpler series of instructions outperforms a single IA-32 instruction
  - MOV$S$
  - LOOP