Computer Organization
CS 231-01

Introducing the Mic-1

Dr. William H. Robinson
October 4, 2004

http://eecs.vanderbilt.edu/courses/cs231/

Opportunity for Easy Points

• Prepare for Quiz #2

Topics

“Two roads diverged in a wood, and I--
I took the one less traveled by,
and that has made all the difference”

– Robert Frost

• Datapath components
• Datapath timing
• Microinstructions

• Announcements
  – Homework #2 is posted on the website
  – Dr. Robinson will be traveling Friday, Oct. 8

Guest Lecture

• Dr. Robinson will be on travel
  – Friday, October 8, 2004

• Dr. Larry Dowdy will cover the lecture
  – Lecture topics will remain on schedule
  – Responsible for material presented
Mic-1 Simulator for Class

Java-based simulator which implements the Mic-1 microarchitecture described in Chapter 4
• http://www.ontko.com/mic1/

Programming assignments will use this!!!

Microarchitecture Level

Level 2
Instruction set architecture level
- Interpretation (microprogram) or direct execution

Level 1
Microarchitecture level
- Hardware

Level 0
Digital logic level

- Purpose: implement the ISA level above it
- Use digital logic “building blocks” as foundation

Block Diagram of Microarchitecture (Mic-1)

Datapath
Part of CPU containing ALU, its inputs, and its outputs

Purpose
Implement the ISA level above it (macro-architecture)

Control Section
Part of CPU containing the H/W necessary to direct the datapath

Datapath of IJVM

Example ISA: IJVM
- Subset of Java Virtual Machine (JVM)
- Only integer instructions

Example microarchitecture
- 32-bit architecture (4 bytes/word)
- Microprogram (in ROM)
- Loop to fetch, decode, and execute instructions
Datapath Buses

- **A Bus**
  - Only connected to H register
  - Becomes the left input of the ALU

- **B Bus**
  - Connected to all other registers except MAR
  - Becomes the right input to the ALU
  - Only want one register enabled

- **C Bus**
  - Connected to all registers
  - Writes output to all enabled registers

Datapath Registers

- Only accessible at the microarchitecture level by the microprogram
- Maintains the state of the computer
- Usually hold an address or a data value corresponding to an ISA variable of the same name

Datapath Registers (Memory Control)

- **MAR**
  - Memory Address Register
  - Contains word addresses for memory references
- **MDR**
  - Memory Data Register
  - Contains data words to/from memory references
- **PC**
  - Program Counter
  - Points to next instruction to be executed
- **MBR**
  - Memory Buffer Register
  - 8-bit register used to read one byte from memory (signed or unsigned)
Datapath Registers (Stack)

- **SP**
  - Stack Pointer
  - Contains the address of the highest word in the local variable frame
- **LV**
  - Local Variable pointer
  - Contains the address of the first location in the local variable frame
- **CPP**
  - Constant Pool Pointer
  - Contains the address of the first word of the constant pool
- **TOS**
  - Top Of Stack
  - Contains the value of highest word on memory stack

Datapath Registers (Miscellaneous)

- **OPC**
  - Couldn’t find what it stood for!
  - A temporary (scratch) register without a preassigned use
- **H**
  - Holding register
  - Attached to the left input of the ALU

Datapath Control Signals

- **B Bus Enable**
  - Enable one register onto B bus
- **C Bus Enable**
  - Write C bus into register(s)
- **ALU Control**
  - Six signals from Fig 4-2
  - What are N and Z from the ALU?
  - Status flags used to determine next microinstruction
- **Shifter Control** (p. 207)
  - Shift Left Logical 8
  - Shift Right Arithmetic 1

Can I read and write the same register in one cycle?

Datapath Timing

- **Has a finite propagation time**
  - Signals travel along wires, through transistors, etc.
- **Implicit clock subcycles**
  - Set up control signals to drive datapath
  - Register loaded onto B bus
  - ALU and shifter operate
  - Result propagates along C bus to registers
Clocks

- Circuit that emits a series of pulses
  - Precise pulse width
  - Precise interval between consecutive pulses
    - Clock cycle time – time interval between corresponding edges of two consecutive pulses

- Events can be triggered by

  ![Clock Cycle Time Diagram](image)

Generation of Asymmetric Clock

- \( C_1 \) and \( C_2 \) are \text{AND}ed together to form signal \( C \)

Timing Diagram for One Datapath Cycle

- Cycle 1 starts here
- Shifter output stable
- Registers loaded instantaneously from C bus and memory on rising edge of clock
- Clock cycle 1
- ALU and shifter
- MPC available here
- New MPC used to load MIR with next microinstruction here
- Set up signals to drive data path
- Drive H and B bus
- Propagation from shifter to registers
- Clock cycle 2

Single-Stage Datapath

- Making the design work requires:
  - Rigid timing
  - Long clock cycle
  - Known minimum propagation time through ALU
  - Fast load of registers from C Bus

- ALU runs continuously
  - Inputs and outputs aren't valid until right "subcycle"

- Only explicit signals driving datapath
  - Falling edge and rising edge of clock