ECE 2300
Digital Logic & Computer Organization
Spring 2021

Single Cycle Microprocessor
Announcements

• How to earn a bonus point (= 1 extra slip day on any of the remaining assignments)
  – Answer bonus question during lectures
  – Watch the recorded Verilog tutorial and provide a short review (via email to ece2300-staff)

• Prelab 3B and Lab 3A due soon
  – Check your FSM against Prelab 3A solution
  – Read the latest Piazza post on timing violation
Review: SRAM vs. DRAM

- **SRAM**: usually on the same chip with CPU
  (+) Fast access
  (−) Relatively high area & cost per bit (6T)

- **DRAM**: typically off-chip; used for main memory
  (+) Single transistor bit cell (1T)
  • Lower area & lower cost per bit
  (−) Slow: need periodic refresh
Course Roadmap (Part 1)

- Boolean algebra
- Combinational logic and minimization
- Logic functions
- CMOS gates
- Binary arithmetic and ALUs
- Latches and flip-flops
- Counters
- Verilog
- Finite state machines
- Hazards, timing, clocking
- Memories
Part 2: Computer Organization

Organization of a Computer

- Processor
- Datapath
- Control
- Memory
- Input
- Output
Let’s Build a Microprocessor!
The Basic Processing Cycle

- Read data from two registers
- Perform an operation
- Place the result into a register

- All three steps performed in 1 clock cycle
Register File (RF)

• **Collection of** $2^k$ n-bit registers

• **Data outputs** (two read ports)
  - DataA – Output data A
  - DataB – Output data B

• **Data inputs** (one write port)
  - D_in – Input data

• **Control inputs**
  - SA – Source address A
  - SB – Source address B
  - DR – Destination address
  - LD – Load destination register with D_in (i.e., write enable)
Example RF Organization

Example with 4 registers. Typically have 32 or more.
Instruction Execution

ADD R0, R1, R2  // R0 <= R1 + R2

operation
destination register
source registers
Instruction Execution

ADD R0, R1, R2

CLOCK
Operation
SA
SB
DataA
DataB
Data Out
DR
LD

“add”
01
10
[R1]
[R2]
[R1]+[R2]
00
Instruction Execution

ADD R0, R1, R2
SUB R3, R2, R1

[CLOCK]

<table>
<thead>
<tr>
<th>Operation</th>
<th>“add”</th>
<th>“sub”</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>01</td>
<td>10</td>
</tr>
<tr>
<td>SB</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>DataA</td>
<td>[R1]</td>
<td>[R2]</td>
</tr>
<tr>
<td>DataB</td>
<td>[R2]</td>
<td>[R1]</td>
</tr>
<tr>
<td>Data Out</td>
<td>[R1]+[R2]</td>
<td>[R2]−[R1]</td>
</tr>
<tr>
<td>DR</td>
<td>00</td>
<td>11</td>
</tr>
<tr>
<td>LD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[R2] means value stored in register R2
Operations With Constants

- Constants are called *immediate values* (IMM)
- Sign extend (SE) IMM to the width of DataA to perform correct two’s complement operation
  - Why? not enough bits in instruction (explained later)
  - Assume IMM is 4 bits and DataA is 8 bits wide
    
    | IMM | SE | Operation | Data Out |
    |-----|----|-----------|---------|
    | 0001 | 0  | +1        | 00000000 |
    | 1110 | 1  | -1        | 11111110 |

ADDI R3, R3, 1
// R3 <= R3 + 1
• Most data are held in memory (RAM)
• Must be moved into a register in order to operate on it
• Data will also move out of registers into memory
  – To make room for other data
  – Or to later move it to permanent storage (e.g., disk)
Reading from Memory ("Load")

**Example:** LOAD R3, 4(R1)  // R3 <= M[R1 + 4]

**Step 1:** Form the memory address by adding the value in R1 (base) with the immediate 4 (offset)

**Step 2:** Read the data at that address in RAM and place it in R3
Writing to Memory ("Store")

Example: \texttt{STORE R2, 0(R0)} \quad // M[R0] <= R2

**Step 1:** Form the memory address by adding the value in R0 with the immediate 0

**Step 2:** Write the value in R2 into the RAM at that address
Control Unit (CU)

• Regulates the interaction between data and operations on data (i.e., datapath)

• Series of control words control the datapath to perform a sequence of operations

• The sequence of operations performed by the CU may be affected by the ALU Condition Codes
  – Z: Zero
  – N: Negative
  – Also V: Overflow and C: Carry out
Datapath + Control Unit

Control Word

- k-bit register addresses
- immediate value
- function select
- register/immediate select
- ALU/memory select
- memory write
- load register
### Sequence of Operations

**Assuming 8 registers in the RF**

```
R2 <= R0 + R1
R1 <= M[R2]
M[R2] <= R0
```

<table>
<thead>
<tr>
<th></th>
<th>DR</th>
<th>SA</th>
<th>SB</th>
<th>IMM</th>
<th>MB</th>
<th>FS</th>
<th>MD</th>
<th>LD</th>
<th>MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>010</td>
<td>000</td>
<td>001</td>
<td>x</td>
<td>0</td>
<td>ADD</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>001</td>
<td>010</td>
<td>x</td>
<td>0</td>
<td>1</td>
<td>ADD</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>x</td>
<td>010</td>
<td>000</td>
<td>0</td>
<td>1</td>
<td>ADD</td>
<td>x</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

---

Lecture 15: 19
Next Time

More Single-Cycle Microprocessor