Review

- As binary, what is \(1111\ 1111\)
  \[\begin{array}{c}
  + \\
  \hline
  1
  \end{array}\]
Suppose I Want Text On My Canvas

- Start by checking the Processing Reference

```
// The font must be located in the sketch's
// "data" directory to load successfully
font = loadFont("FFScala-32.vlw");
textFont(font);
text("word", 15, 30);
fill(0, 102, 153);
text("word", 15, 60);
fill(0, 102, 153, 51);
text("word", 15, 90);
```

- What do you notice from their example?
Need A Font For Your Project

- Create a Font From Under Tools
Pick One

- Pick a font; remember its name and size!
What Needs To Go In Your Code?

- Check the steps in the example

```
// The font must be located in the sketch's
// "data" directory to load successfully
pFont font;
font = loadFont("FSScala-32.vlw");
textFont(font);
text("word", 15, 30);
fill(0, 102, 153);
text("word", 15, 60);
fill(0, 102, 153, 51);
text("word", 15, 90);
```

- loadFont(); announce font with textFont(); use
Try It On A Tiny Example

text below figure

```java
PFont font;
void setup() {
    size(300,300);
    background(0);
    noStroke();
    font = loadFont("Serif-48.vlw");
    textFont(font);
}
void draw() {
    fill(255,0,0);
    text("text below figure", 20, 100);
    fill(255);
    rect(mouseX, mouseY, 40,40);
}
```
Instruction Execution is ... Simple, Even A Computer Can Do It

Lawrence Snyder
University of Washington, Seattle
Our first discussion of Lightbot noted that the instructions were formed of composite operations ...

Today ... we see that computer’s instructions are, too
Computers ...

- Deterministically execute instructions to process information

  “Deterministically” means that when a computer chooses the next instruction to perform it is required by its construction to execute a specific instruction based only on the program and input it is given.

Computers have no free will and they are not cruel.
Fetch/Execute Cycle

- Computer = instruction execution engine
  - The fetch/execute cycle is the process that executes instructions
    - Instruction Fetch (IF)
    - Instruction Decode (ID)
    - Data Fetch (DF)
    - Instruction Execution (EX)
    - Result Return (RR)

- The computer internal parts implement this cycle
Anatomy of a Computer

The Hard Disk is the $\alpha$-device
Memory ...

- Programs and their data must be in the memory while they are running.

Groups of four bytes are a word.

Memory contents:

01000100
Fetch/Execute cycle is hardwired in computer’s control; it’s the “engine”

The instructions executed have the form
ADDB 20, 10, 16

Put in memory location 20 the contents of memory location 10 + contents of memory location 16
### Indirect Data Reference

- Instructions tell *where* the data is, not *what* the data is ... contents change

One instruction has many effects

```
ADDB 20, 10, 16
```

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-55</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Arithmetic/Logic Unit does the actual computing.

Each type of data has its own separate instructions:
- ADDB: add bytes
- ADDBU: add bytes unsigned
- ADDH: add half words
- ADDHU: add halves unsigned
- ADD: add words
- ADDU: add words unsigned
- ADDS: add short decimal numbers
- ADDD: add long decimal numbers

Most computers have only about a 100-150 instructions hard wired.
Input units bring data to memory from outside world; output units send data to outside world from memory

- Most peripheral devices are “dumb” meaning that the processor assists in their operation
- Disks are *memory* devices because they can output information and input it back again
The PC’s PC

- The program counter (PC) tells where the next instruction comes from
  - Instructions are a word long, so add 4 to the PC to find the next instruction

Program Counter:  112

688,724  ADD 210,216,220  AND 414,418,720  OR ...
Run Instruction: 2200: Add 800, 428, 884
Instruction Fetch: Get Some Work

- **ALU**
- **Control**
  - PC: 2200
  - ADD 800,428,884

- **Memory**
  - 2200: ADD 800,428,884

- **Input**

- **Output**
Instruction Decode: What To Do?

ALU

[428]
[884]
[800]

Control

ADD 800,428,884

Memory

Input

Output
Data Fetch: What’s The Input

ALU

[428]  42
[884]  +  12
[800]  -----

Control

PC: 2204

Memory

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>428</td>
<td></td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>884</td>
<td>12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Input

Output
Instruction Execution: Just Do It

- **ALU**:
  - 42
  - 12
  - 800

- **Control**:
  - PC: 2204

- **Input**

- **Memory**

- **Output**
Result Return: Put It Away 4 Future
Clocks Run The Engine

- The rate a computer “spins around” the Fetch/Execute cycle is controlled by its clock
  - Current clocks run 2-3 GHz
  - In principle, the computer should do one instruction per cycle, but often it fails to
  - Modern processors try to do more than one instruction per cycle, and often succeed

**Clock rate is not a good indicator of speed**
Jack Kilby, Mr. Integrated Circuits
Semiconductors

- Silicon, a semiconductor -- sometimes it conducts and sometimes it doesn’t
  - It’s possible to control when semiconductors do and don’t conduct

**Compute by controlling conducting**

Ex.: AND 428, 884, 800

- Make semiconductor conduct if memory location 428 is true
- Send “yes” signal on wire
- Make semiconductor conduct if memory location 884 is true
- Detect presence/absence of “yes”
Field Effect

- Charged objects are familiar -- use a nylon comb on a dry day
  - A charged field can control whether
  - a semiconductor conducts or not

The charge of the control wire (gate) is key
- Neutral gate, channel doesn’t conduct
- Charged gate, channel conducts
MOS Transistors

- The field-effect idea is implemented in metal-oxide-semiconductor transistors
- Schematic in Si
nMOS Transistor
Summary

- Fetch/execute cycle runs instructions
  - 5 steps to interpret machine instructions
  - Programs must be in the memory
  - Data is moved in and out of memory

Instructions, data are represented in binary