What We've Learned About TOY

TOY machine:
- Box with switches and lights.
- 16-bit memory locations, 16-bit registers, 8-bit pc.
- 4,328 bits = \((255 \times 16) + (15 \times 16) + (8)\) = 541 bytes!
- von Neumann architecture.

TOY programming:
- TOY instruction set architecture: 16 instruction types.
- Variables, arithmetic, loops.

What We Do Today

Data representation. Negative numbers.
Input and output. Standard input, standard output.
Manipulate addresses. References (pointers) and arrays.
TOY simulator in Java.

Data Representation

Data is a sequence of bits (interpreted in different ways)
- Integers, real numbers, characters, strings,
- Documents, pictures, sounds, movies, Java programs, ...

Ex. 01110101
- As binary integer: \(1 + 4 + 16 + 32 + 64 = 117\)\(_{10}\)
- As character: 117th Unicode character = 'u'.
- As music: 117/256 position of speaker.
- As grayscale value: 45.7% black.

Adding and Subtracting Binary Numbers

Decimal and binary addition.

```
  011  1  1
+ 0 0 1  0 1
---
  0 1 0 1 0
```

Subtraction. Add a negative integer

```
  0 1 0 1 1 0 0
+ 0 1 1 0 1 0 1
---
  1 1 0 0 0 1 1
```

Q. How to represent negative integers?
We could use 16 bits to represent 0 to 2^16 - 1.
1. We want negative integers too.
2. Reserving half the possible bit-patterns for negative seems fair.

Highly desirable property: If x is an integer, then the representation of -x, when added to x, is zero.

```
\[
\begin{array}{c|c}
\text{Original} & \text{Reversed} \\
\hline
0 & 0000000000000000 \\
1 & 1111111111111111 \\
\end{array}
\]
```

Addition and subtraction are easy.

Reserving half the possible bit-patterns for negative seems fair.

Two’s Complement Integers

To compute -x from x:
1. Start with x.
2. Flip bits.
3. Add one.

```
\begin{array}{c|c}
\text{Original} & \text{Reversed} \\
\hline
0 & 0000000000000000 \\
1 & 1111111111111111 \\
\end{array}
```

Properties of Two’s Complement Integers

- Leading bit (bit 15) signifies sign.
- Addition and subtraction are easy.
- Checking for arithmetic overflow is easy.
- Negative integer -x represented by 2^16 - x.
- Not symmetric: can represent -32,768 but not 32,768.

Java: Java’s int data type is a 32-bit two’s complement integer.

Ex. 2,147,483,647 + 1 equals -2,147,483,648.

Representing Other Primitive Data Types in TOY

- **Bigger integers**: Use two 16-bit TOY words per 32-bit Java int.
- **Real numbers**: Use IEEE floating point (like scientific notation).
- **Characters**: Use one 16-bit TOY word per 16-bit Java Unicode char.

Note: Real microprocessors add hardware support for int and double.
Standard Output

Writing to memory location $FF$ sends one word to TOY stdout.

Ex. `$9AFF` writes the integer in register $A$ to stdout.

Standard Input

Loading from memory address $FF$ loads one word from TOY stdin.

Ex. `$8AFF` reads an integer from stdin and store it in register $A$.

Standard Input and Output: Implications

Standard input and output enable you to:

- Get information out of machine.
- Put information from real world into machine.
- Process more information than fits in memory.
- Interact with the computer while it is running.

Arrays in TOY

TOY main memory is a giant array.

- Can access memory cell 30 using load and store.
- Goal: access memory cell where $i$ is a variable.

Load indirect. [opcode A]

- `AC06` means load $mem[i]$ into register $C$.
- `$i$ is variable index.

Store indirect. [opcode B]

- `AC06` stores content of register $C$ into $mem[i]$.

Arrays in TOY

TOY main memory is a giant array.

- Can access memory cell 30 using load and store.
- Goal: access memory cell where $i$ is a variable.

Load indirect. [opcode A]

- `AC06` means load $mem[i]$ into register $C$.
- `$i$ is variable index.

Store indirect. [opcode B]

- `AC06` stores content of register $C$ into $mem[i]$.

for int $i = 0; i < N; i++$

```java
a[i] = StdIn.readInt();
StdOut.println(a[N-i-1]);
```
TOY Implementation of Reverse

TOY implementation of reverse.

- Read in a sequence of integers and store in memory 30, 31, 32, ... until reading 0000.
- Print sequence in reverse order.

```plaintext
10: 7101 R1 ← 0001 constant 1
11: 7A00 RA ← 0300 a[]
12: 7B00 RB ← 0000 n
13: ECFF read RC
14: CD19 if (RC == 0) goto 19
15: 16AB R6 ← RA + RB
16: BC06 mem[R6] ← RC a[n]
17: 1BB1 RB ← RB + R1 n++
18: C013 goto 13
```

read in the data

```plaintext
19: CB20 if (RB == 0) goto 20
1A: 16AB R6 ← RA + RB
1B: 2661 R6 ← R6 - R1
1C: AC06 RC ← mem[R6] a[n-1]
1D: 9CFF write RC
1E: 2BB1 RB ← RB - R1 n--
1F: C019 goto 19
```

print in reverse order

Unsafe Code at any Speed

Q. What happens if we make array start at 00 instead of 30?
A. Self modifying program; can overflow buffer and run arbitrary code.

```plaintext
10: 7101 R1 ← 0001 constant 1
11: 7A00 RA ← 0000 a[]
12: 7B00 RB ← 0000 n
13: ECFF read RC
14: CD19 if (RC == 0) goto 19
15: 16AB R6 ← RA + RB
16: BC06 mem[R6] ← RC a[n]
17: 1BB1 RB ← RB + R1 n++
18: C013 goto 13
```

Buffer Overflow Attacks

Stuxnet worm. [July 2010]

- Step 1. Natanz centrifuge fuel-refining plant employee plugs in USB flash drive.
- Step 2. Data becomes code by exploiting Window buffer overflow; machine is owned.
- Step 3. Uranium enrichment in Iran stalled.

More buffer overflow attacks: Morris worm, Code Red, SQL Slammer, iPhone unlocking, Xbox softmod, JPEG of death, ...

Lesson.

- Not easy to write error-free software.
- Embrace Java security features.
- Keep your OS patched.

Buffer Overflow Example: JPEG of Death

Microsoft Windows JPEG bug. [September, 2004]

- Step 1. User views malicious JPEG in IE or Outlook.
- Step 2. Machine is owned.
- Step 3. Uranium enrichment in Iran stalled.

Lesson.

- Not easy to write error-free software.
- Embrace Java security features.
- Don’t try to maintain several copies of the same file.
- Keep your OS patched.

What Can Happen When We Lose Control (in C or C++)?

Buffer overflow.

- Array buffer[] has size 100.
- User might enter 200 characters.
- Might lose control of machine behavior.

Consequences.

Viruses and worms.

Java enforces security.

- Type safety.
- Array bounds checking.
- Not foolproof.

Buffer Overflow Example: JPEG of Death

Microsoft Windows JPEG bug. [September, 2004]

- Step 1. User views malicious JPEG in IE or Outlook.
- Step 2. Machine is owned.
- Step 3. Uranium enrichment in Iran stalled.

Lesson.

- Not easy to write error-free software.
- Embrace Java security features.
- Don’t try to maintain several copies of the same file.
- Keep your OS patched.
Dumping

Q. Work all day to develop operating system in mem[10] to mem[FF]. How to save it?
A. Write short program dump.toy and run it to dump contents of memory onto tape.

Booting

Q. How do you get it back?
A. Write short program boot.toy and run it to read contents of mem[10] to mem[FF] from tape.

Extra Slides

Two’s Complement Arithmetic

Addition is carried out as if all integers were positive.
- It usually works.
- But overflow can occur.

Java and TOY

Correspondence between Java constructs and TOY mechanisms:

Two’s Complement Arithmetic

Addition is carried out as if all integers were positive.