Eric Roberts
CSCI 121

Handout #18
September 24, 2018

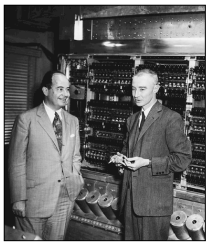
Stored-Program Machines

Stored Program Machines

Eric Roberts
CSCI 121
September 24, 2018

The von Neumann Architecture

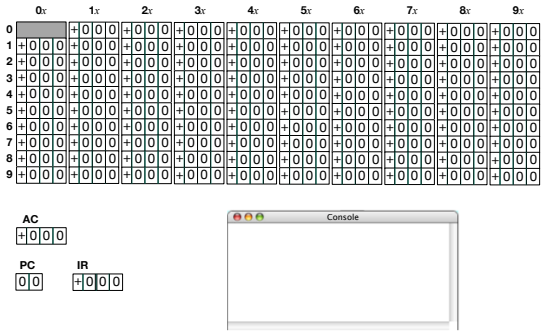
- One of the foundational ideas of modern computing—traditionally attributed to John von Neumann although others can make valid claims to the idea—is that code is stored in the same memory as data. This concept is called the ***stored programming model***.
- My story today consists of two film clips about the Manchester Baby, which was the first stored-program computer. For the rest of the class, I will describe a slightly more powerful machine that I've nicknamed *Toddler*.



John von Neumann and J. Robert Oppenheimer

The Manchester Baby

Structure of the Toddler Machine



The Toddler Instruction Set

1*xx*	**LOAD** *xx*	Loads the value from address *xx* into the AC
2*xx*	**STORE** *xx*	Stores the value from AC into address *xx*
3*xx*	**ADD** *xx*	Adds the value at address *xx* to the AC
4*xx*	**SUB** *xx*	Subtracts the value at address *xx* from AC
500	**HALT**	Halts the machine
8*xx*	**INPUT** *xx*	Reads a value into address *xx*
9*xx*	**OUTPUT** *xx*	Prints the value in address *xx*

The Add-Two-Numbers Program

```
(01) +850    INPUT 50
(02) +851    INPUT 51
(03) +150    LOAD 50
(04) +351    ADD 51
(05) +252    STORE 52
(06) +952    OUTPUT 52
(07) +500    HALT
```

The Instruction Cycle

1. *Fetch the current instruction.* In this phase, Toddler finds the word from the memory address specified by the **PC** and copies its value into the IR.
2. *Increment the program counter.* Once the current instruction has been copied into the IR, Toddler adds one to the PC so that it points to the next instruction.
3. *Decode the instruction in the instruction register.* The value copied into the IR is a three-digit integer. To use it as an instruction, Toddler must divide the instruction word into its *opcode* and *address* components.
4. *Execute the instruction.* Once the operation code and address field have been identified, the Toddler processor must carry out the steps necessary to perform the indicated action.

The Toddler Instruction Set

1*xx*	**LOAD** *xx*	Loads the value from address *xx* into the AC
2*xx*	**STORE** *xx*	Stores the value from AC into address *xx*
3*xx*	**ADD** *xx*	Adds the value at address *xx* to the AC
4*xx*	**SUB** *xx*	Subtracts the value at address *xx* from AC
500	**HALT**	Halts the machine
5*xx*	**JUMP** *xx*	Takes the next instruction from address *xx*
6*xx*	**JUMPZ** *xx*	Jumps to *xx* if the AC is zero
7*xx*	**JUMPN** *xx*	Jumps to *xx* if the AC is negative
8*xx*	**INPUT** *xx*	Reads a value into address *xx*
9*xx*	**OUTPUT** *xx*	Prints the value in address *xx*

The Countdown Program

assembly language

```
(01) +111   start:  LOAD ten
(02) +212           STORE i
(03) +709   loop:   JUMPN done
(04) +912           OUTPUT i
(05) +112           LOAD i
(06) +410           SUB one
(07) +212           STORE i
(08) +503           JUMP loop
(09) +500   done:   HALT
(10) +001   one:    1
(11) +010   ten:    10
(12) +000   i:      0
```

Representing Constants

- Constants in the Toddler machine need to be stored in one of the memory addresses, as illustrated by the following lines from the assembly language version of **Countdown.td**:

```
one:    1
ten:    10
```

- The instruction **LOAD ten** then refers to a memory address that contains the value 10.
- Toddler also allows you to write

```
LOAD #10
```

which finds space for the constant 10 at the end of the program and then fills in the **LOAD** instruction with the address of that constant.

Exercise: Multiply Two Numbers

- How would you write a Toddler program to multiply two nonnegative numbers, even though the machine has no multiply instruction?

How Is Toddler Useful?

- Toddler makes it easier for you to imagine how the computer carries out the steps in a Python program.
- The idea of memory addresses is critical to understanding the concept of ***references***, which are implemented internally by storing the address of a data value in memory.
- All objects in Python are stored as references, which means that those objects can be shared. This behavior is illustrated in both Chapter 3 and Handout #17.
- Handout #17 also notes that Python stores the characters in a string in consecutive memory locations. This model is easy to visualize in the Toddler environment.