Introduction to Computer Science

Presented by
Don Stokes
Team 3641

April 29, 2015

Binary Numbers

- Humans use a base 10 numbering system.
  - Each digit can take on 10 different values (0 … 9).
  - Subsequent digits are worth 10 times the previous digit.
- Computers use a base 2 numbering system.
  - Each digit can take on the values 0 or 1.
  - Subsequent digits are worth 2 times the previous digit.
- Example: Binary number 1011
  - Decimal equivalent is 1*8+0*4+1*2+1*1 = 11
- Computers use binary because values can be represented by electrical switches that are off or on with no ambiguity.

Binary Addition

- 0 + 0 = 0; 0 + 1 = 1; 1 + 1 = 10 (with a carry bit)
- Four bit example:

```
   1010
+  0011
=  1101
```

Complement Operation

- One's Complement is calculated by flipping all bits.
- Example: One's Complement of 0011 is 1100.

- Two's Complement is One's Complement plus 1.
- Example: Two's Complement of 0011 is 1101.
- Negative binary numbers are expressed as the Two's Complement of the positive number.
- Example:
  Decimal -7
  Binary -0111 = 1000 + 1 = 1001

Binary Subtraction

- Binary subtraction can be done by adding a negative value (Two's Complement).
- Example:
  Decimal: 10 – 7 = 3
  Binary: 1010 + (1000 + 1) = 0011
  (notice the carry from the fourth bit is lost)

Hexadecimal

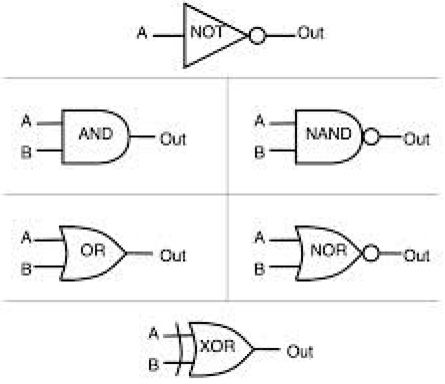
- It is not easy to convert large numbers between binary and decimal. Try it! The problem is due to 10 not being a power of 2.
- Binary numbers can be cumbersome to deal with. We need 16 digits just to express decimal 64,000.
- For convenience, we often use a base 16 numbering system called *hexadecimal*. Each hexadecimal digit will align with each 4 binary digits. Converting to/from binary never requires dealing with a number larger than 15 decimal.
- Hex Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Example: Binary 10110100 = Hex B4
  - Decimal: 11*16+4=180
- Sometimes a base 8 system called *octal* is used.

Boolean Operators

- Boolean logic is math based on values FALSE and TRUE.
  - (or 0 and 1)
- Base 10 math has operators like addition, subtraction, multiplication and division.
- Binary operators are NOT, AND, OR, XOR.
  - NOT: Result is the opposite of the operand.
  - AND: Result is 1 if ALL operands are 1, otherwise 0.
  - OR: Result is 1 if ANY operands are 1, otherwise 0.
  - XOR: Result is 1 if an ODD number of operands are 1, otherwise 0.

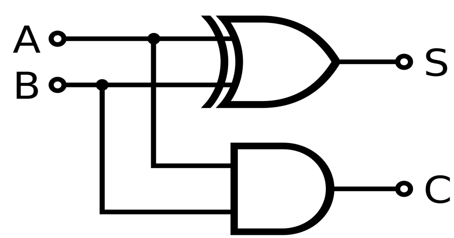
Logic Gates

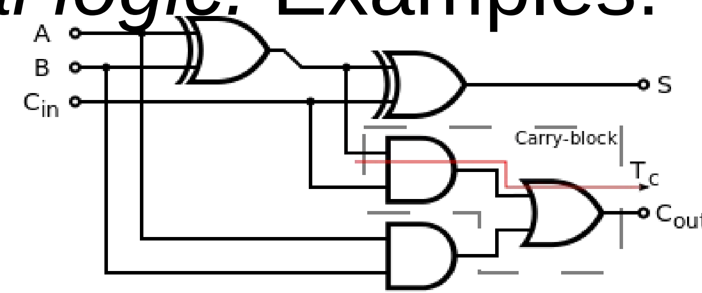
- Electrical circuits designed for boolean logic are called Logic Gates.
- NOT, AND, OR, XOR, NAND, NOR, …



Combinational Logic

- Interesting logic circuits can be constructed from logic gates.
- Logic circuits that only depend upon their current inputs are called *combinational logic.* Examples:
  - Half Adder
  - Full Adder

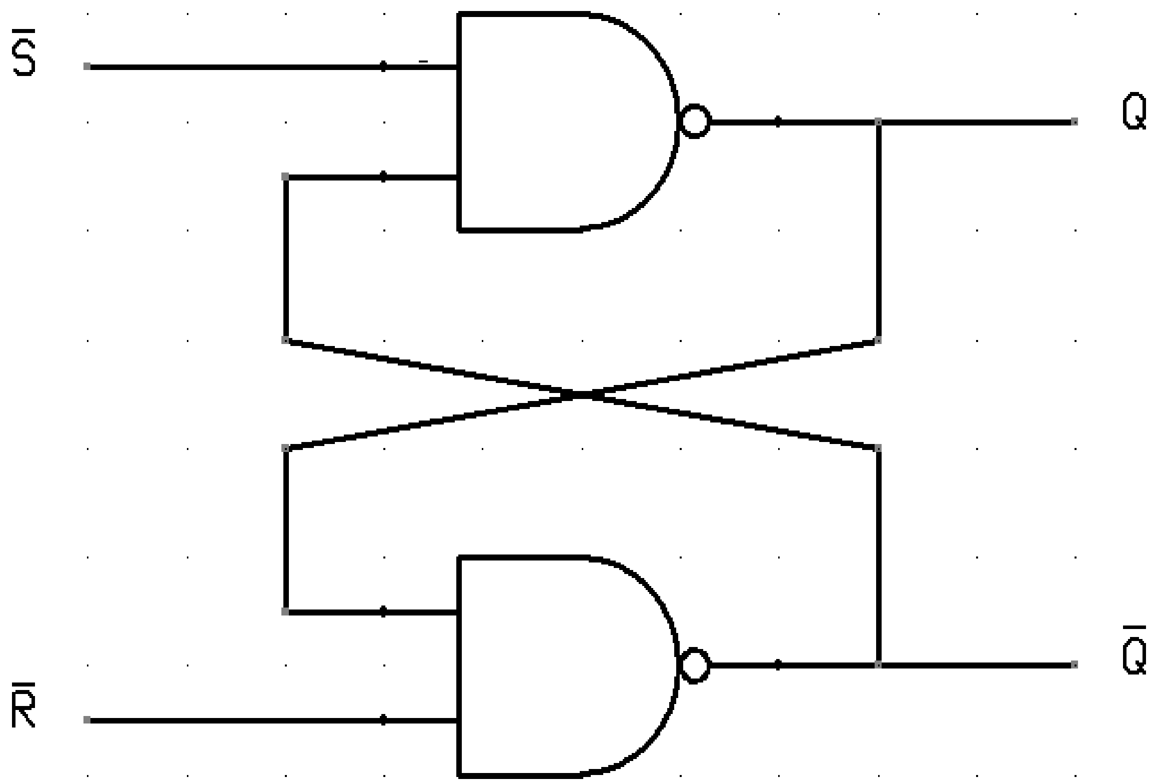




- The output will appear after some *propagation delay*.

Sequential Logic

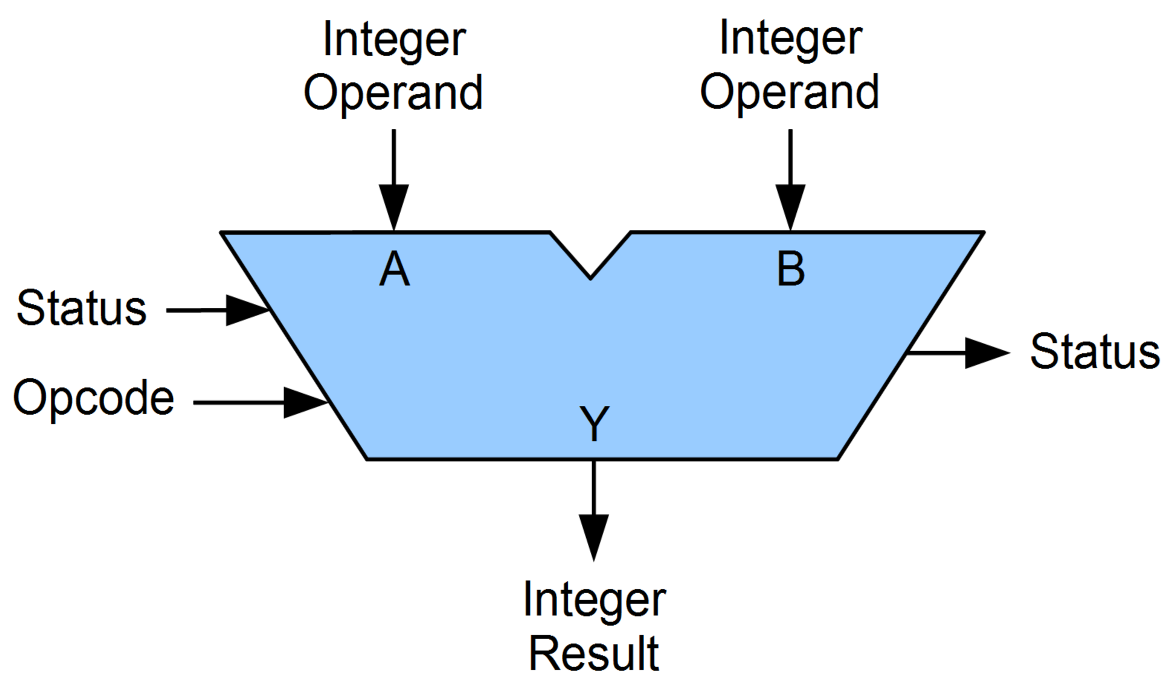
- Logic circuits that depend on their current inputs and previous state are called *sequential logic*.
- Examples:
  - Latches
  - Flip-Flops
  - Counters
  - Registers



- Sequential logic is useful for holding binary values

Arithmetic Logic Unit

- A combinational logic circuit known as an Arithmetic Logic Unit (ALU) is available for computing simple integer math functions like ADD, SUBTRACT, AND, OR, XOR, and Complement.



Memory

- Circuits for storing large amounts of data are called *memory.*
- Memory that cannot be modified once the initial contents are stored is called Read Only Memory (ROM). ROM retains its contents even after power is removed.
- Memory that can read or update any location is called Random Access Memory (RAM). RAM loses its contents after power is removed.

Central Processing Unit

- A sequential logic circuit that executes a list of instructions is a central processing unit (CPU).
- This circuit includes other circuits such as:
  - Registers
  - ALU
  - Instruction Decoding
  - Sequencer
- External circuits are required for clock signal, memory, Input / Output (I/O).
- A clock signal is a periodic pulse that is used for sequencing sequential circuits.

Microcontroller

- A microcontroller (MCU) is a circuit that combines a CPU, clock, memory, and I/O peripherals.
- An MCU is typically embedded in a product for control purposes.
- One product example is ROBOTS!

System on a Chip (SOC)

- An SOC is similar to an MCU, but it contains very sophisticated peripherals, such as USB and graphics display controllers.
- By combining an SOC and a few components, a complete computer can be produced. Examples: Raspberry Pi, Beagle Bone

A Simple Microprocessor

- Intel 8080 (from 1974)
- 8 Bit Microprocessor
- Useful Introduction: http://en.wikipedia.org/wiki/Intel_8080
- Download the user manual here: http://www.elenota.pl/datasheet-pdf/133557/Intel/8080
- Play with an emulator here (use Firefox): http://bluishcoder.co.nz/js8080/

Programming

- A CPU repeats a cycle of:
  - Fetching the next instruction from memory (determined by the Program Counter).
  - Advancing the Program Counter
  - Decoding the instruction
  - Modifying registers, memory or peripherals
- The activity of designing, implementing and debugging a set of instructions (program) is called *programming*.

Machine Code Programming

- Programming in the binary language that a CPU understands is called machine code programming.
- Nobody does this anymore because we have computer programs called *assemblers* that automate the process.

Assembly Language Programming

- This is the lowest level of programming done by programmers.
- Each line of an assembly language program usually corresponds to one instruction that the CPU executes.
- Additional lines are programmed to tell the assembler how to generate the machine code.
- Instead of entering the machine code instruction number into the program, human friendly mnemonics are used.
  - Example: ADD is the addition instruction. 3 is the machine code on the PC.
- The assembler will keep track of memory addresses and allow usage of human readable labels for them.
- Assembly language is time consuming and usually only done when a higher level programming language cannot be used due to technical reasons, like CPU initialization.

The “C” Programming Language

- The “C” programming language is a favorite for machine control.
- This language is the next level up from assembly language.
- C program statements are converted into machine code instructions that are directly executable by CPUs. This leads to programs that run very efficiently (fast, small).
- A program that converts “C” language statements to machine code is called a *C Compiler*.

C Program Sample

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    printf(“Hello, World!\n”);
    return 0;
}
```

C Plus Plus (C++)

- The C++ programming language is a superset of the C programming language.
- C++ adds object oriented concepts to C. (C is procedure oriented.)
- C++ programs require a larger runtime library, which results in a bigger program. (Libraries are pre-compiled statements that are combined with the programmer's statements to produce the executable program.)
- Editorial: Do not start learning C++ until you have learned C! You learned to walk before you learned to run.

C++ Program Sample

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[]) {
    cout << "Hello, World!" << endl;
    return 0;
}
```

High Level Languages

- High level languages are easier to learn than lower level languages. (auto type conversion, no declarations, ...)
- Some high level languages include simple statements for doing complex operations, like manipulating databases (SQL).
- Many high level languages do not generate machine code and require an *interpreter* program at runtime to execute. The result is slower running programs.
- Examples: Java, BASIC, shell scripts, batch files

Software Development Process

- Analysis: What do we need to do? (gather requirements)
- Design: How will we do it? (planning)
- Implementation: Program according to the design.
- Testing: Execute the program and check results.
- Debugging: Determine cause of failed test.
- Verification and Validation: Are all requirements satisfied?
- Deployment: Deliver software to the user and install.
- Support: Help the user with issues. (training)
- Maintenance: Update the software when new requirements are submitted or problems are found.

UML, SysML

- Unified Modeling Language, System Modeling Language
- Software applications (tools) for analysis and design. Some tools automate some of the implementation. (code generation)
- These are graphical languages for developing pictures that clearly specify the analysis and design.
- Includes diagrams for: Use Cases, Objects/Classes, Object States, Object Collaboration Sequencing, Flow Charts
- A picture is worth a thousand words!

Integrated Development Environment (IDE)

- Software application (tool) for writing programs, compiling executable files, and debugging.
- Programming aids like Auto Completion. It guesses what you want to write and offers you suggestions.
- Advanced searching capabilities. Example: Where are all places a variable is used?
- Popular IDEs: Eclipse, NetBeans, Arduino IDE, Visual Studio

Object Oriented Programming

- Focus is on Classes of Objects
- What properties do objects have?
- What operations can be performed on objects?
- What are the relationships between objects, especially inheritance.
- Details of implementation are hidden by classes.