Computer Science – Lecture 7
Computer Arithmetic

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

Objectives of the Lecture

- To understand how computers process information
- To understand and use numbers in binary and hexadecimal

- Tomorrow's Practical
  - Office tools revision

Homework - 1

- **Apologies, could the following people send their files again, as attachments**
  - **Or use the digital drop box on Blackboard**
    - **Under Tools on left hand side menu**
  - **Tete**
  - **Bodam**
  - **Ghassan**
  - **Olusola**

Homework - 2

- Please hand in your presentations, set up to run automatically

- DO NOT e-mail them if more than 1Mb please!
  - Use Digital Drop Box on Blackboard
  - Under “Tools” in list on left hand side of screen

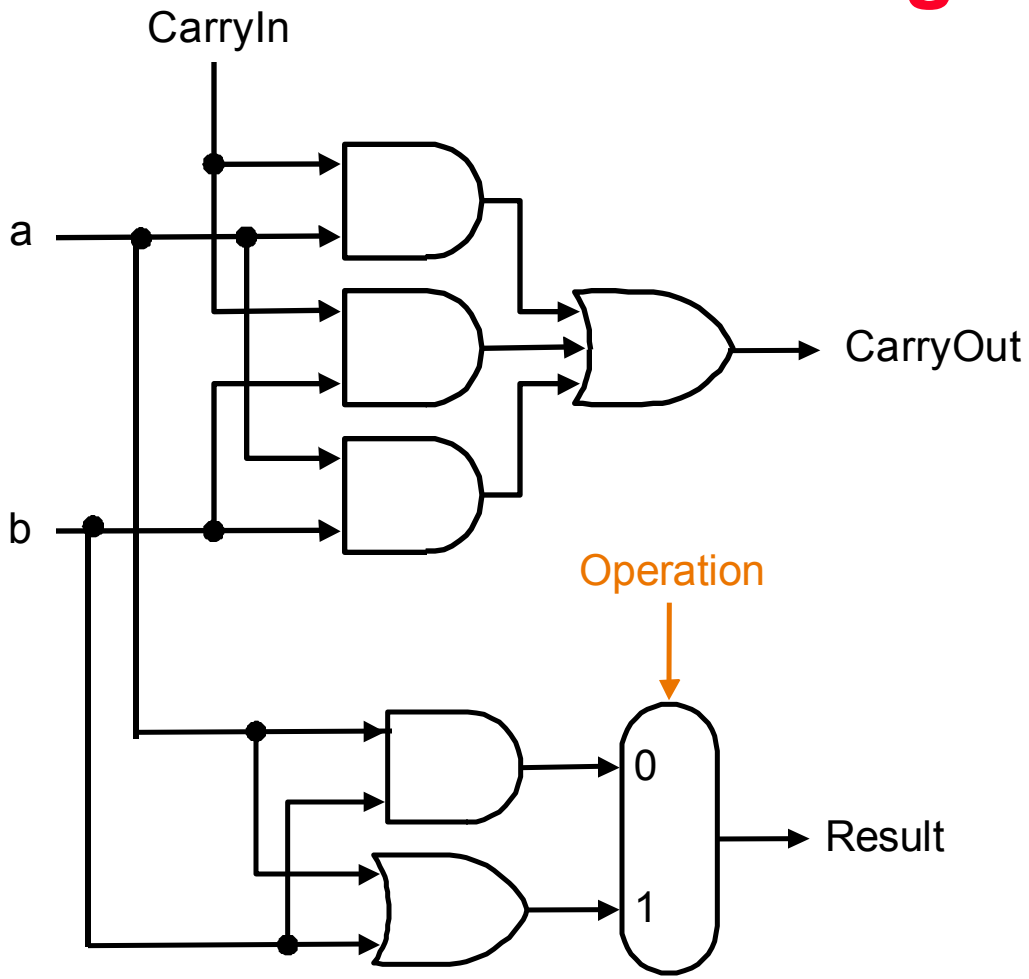
Low Level Computer Hardware

- Recall from our first lecture:
  - Computers work digitally
    - On / off zero / one
  - We built simple logic units
    - AND / OR / NOT
  - We could combine these to carry out higher level functions
    - Majority voting circuit
  - We could combine these into memory elements
    - Flip-flops, storing 1 bit of information

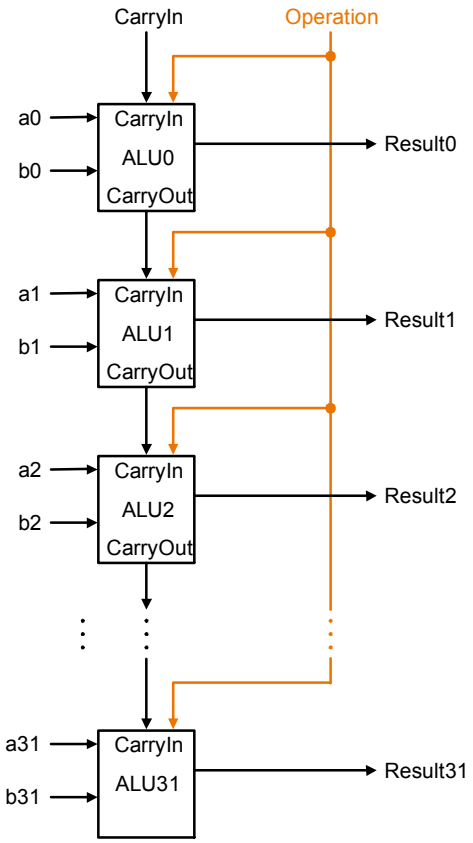
Mid Level Computer Hardware

- We combine these elements to create even higher level functions
  - Logic gates to build an Arithmetic Logic Unit
    - Can add and subtract numbers
  - A row of flip-flops to create a Register
    - Can store larger numbers

A Circuit To Add/subtract Two Digits



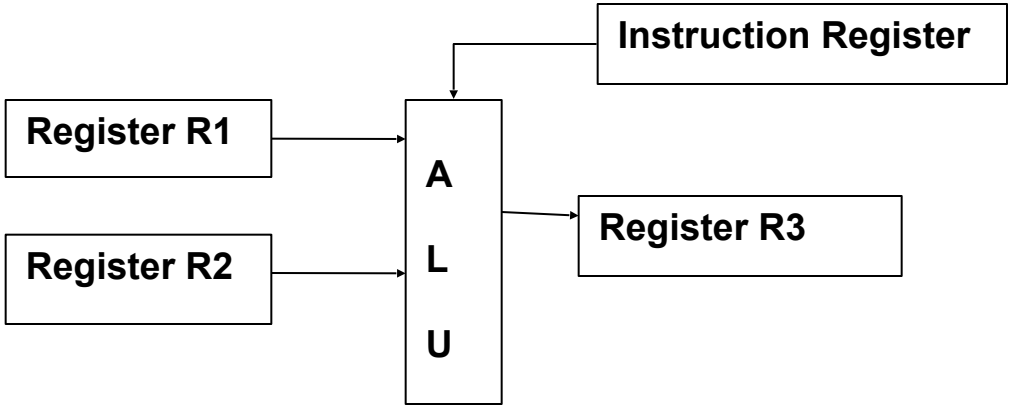
A Multi-bit Adder



The circuit can be repeated to work on any number of bits (typically 8, 16, 32)

The Central Processing Unit

We can combine these further to create a CPU



The instruction register controls what function the ALU should carry out.
Registers can also be loaded to / from main memory

More Information on CPUs

- There is more information on CPUs available on Blackboard
  - Transferring data from registers to main memory
  - More about the instruction register
  - Memory addressing

- You will NOT be examined on this extra material, but most people on the Computer Science degree course will be familiar with it

Number Formats – Bases

- A decimal number (base 10) can be written as:
  - e.g 3241.56_{base10} can be represented as
  - $3 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$

- A base 5 number can be written as:
  - e.g 3241.32_{base5} can be represented as
  - $3 \times 5^3 + 2 \times 5^2 + 4 \times 5^1 + 1 \times 5^0 + 3 \times 5^{-1} + 2 \times 5^{-2} =$ 446.68base10

Binary and Hexadecimal

- A base 2 (binary) number can be written as shown below.
  - e.g 1001_{base2} can be represented as:
  - $1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{base10}$
  - e.g 1001.01_{base2} can be represented as:
  - $1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 9.25_{base10}$
- A base 16 (hexadecimal) number has the following digits:
  - 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - e.g $3F4_{base16}$ can be represented as:
  - $3 \times 16^2 + F \times 16^1 + 4 \times 16^0 = 3 \times 16^2 + 15 \times 16^1 + 4 \times 16^0 = 500_{base10}$

Why Do We Use These?

- Binary
  - Some format as the computer hardware
  - Zero / One
  - On / Off
  - Can see bit patterns
  - 001100110…
- Hexadecimal
  - Easy to convert to binary
  - One hex digit = 4 bits
  - E.g. F4 (hex) =

    1111 0100 (binary)

Conversion To / From Decimal

- See more detailed notes on Blackboard for a full description

  OR

- Use a calculator!

Binary Arithmetic

As for decimal (base 10 numbers)
For binary addition:

```
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10        (10base2 = 2base10)
1 + 1 + 1 = 11    (11base2 = 3base10)

e.g. 36 + 25
          36                  0100100
     +    25             +   0011001
     ----------          ------------------
          61                  0111101
     ----------          ------------------
```

Summary

- The basic operation of a computer is to carry out arithmetic operations on binary numbers stored in registers
- Practical – Review of Office Tools

  PRACTICAL SESSION FROM 1:00 PM IF PREFERRED
- Next week – Foundations of Programming