What is computation?

- Solves some problem
  - Control a device
  - Interface with the user
- Useful manipulation of information
  - Automated execution of an algorithm
- Practical definition
  - Sequence of primitive operations on memory
    - arithmetic and boolean instructions
    - control flow operations
    - Memory (registers, memory, desk??)

How can you express computation?

- Von-Neumann model
  - Variables (memory)
  - If .. Then
  - While …
  - Function calls
  - Modules
  - Encapsulation, all that OO stuff
- Dataflow computing
  - Demand-based execution
- Lambda calculus
- Constraint solving
- Stream-based computation
- Quantum computation
- Cellular computation
- Parallel version thereof
- Neural networks

Registers

- What are
  - On chip memory
  - More is good
  - too much and it gets slow
- Access methods
  - Directly
  - Indexing (far less common)
  - Implicit
- Architectures
  - Accumulator architecture
  - Register based (aka RISC like)
  - Stack

Memory addressing modes

- Direct (often is register indirect via gp)
- Register indirect
  - Register + constant
- Immediate
  - Add r0, #4
- Other ones
  - Memory indirect
  - Etc a = mem[mem[b]]

Instructions

Encodings

- Uniform length
  - Simple hardware
  - Simplify hardware
- Non-uniform length
  - Saves space
  - More efficient?
- No reason to waste space balance against for thought

What else might we want to express?

- Special memory accesses
  - Ports, devices, other processors
- Security
- Management instructions
- Forward looking information
- Explicit parallelism

Instruction Sets – Summary 1

- An instruction set feature (ISF) is beneficial when …
  - Improves performance
  - Makes someone's life easier at not too much expense
  - Benefit out ways cost
  - Easy to compile to
  - Saves space
  - When expresses something useful and it is used

Instruction Set – Summary 2

- An ISF is detrimental when …
  - Slows down the common case
  - Breaks existing code
  - Difficult to implement
  - Difficult to target
  - Narrow sightedness

Instruction Set – Summary 3

- An ISF is *useful* when …
  - When its good and you can target with a compiler

What was 1980 like, for RISC?

- More in the hardware
- Design from discrete components
- Direct support for HLL
- IBM, DEC, Motorola, TI, There was no PC, Zilog,

When is CISC good?

When is RISC good?

Does it matter?

- Yes it does!
  - Extra time, cost, silicon, etc
  - You may pay for it later (I.e. Intel)
- No it does not
  - Economics, we know who won
  - We have plenty of silicon to burn
    - Full employment act for engineers

What is happening now?