16/11/2015

Computer Organisation (cont.)

These tradeoff decisions influence the size of the resulting programs, execution times, ease of programming, etc.

E.g. to do 32-bit arithmetic on an 8-bit machine you have to combine registers and so there are more instructions than if we had a 32-bit register and A.L.U. So the programs are bigger and the execution time is longer.

So we have specialised hardware (processors etc.) for different tasks, e.g. to send someone to Mars (or the Moon).

Special Purpose Registers:

- Program Counter (Instruction Pointer)
  - holds the address of the next instruction to be executed
  - without this we wouldn't know what to execute next.
- Status Register
  - reflects the outcome of the execution of the previous instruction
  - E.g. Did it overflow? Is the result negative? Am I dividing by zero? Is the result 0?
    If I subtract things then I can see they were equal if the result is 0, that the first was bigger if the result was bigger than 0.
- Stack Pointer
  - used to implement subroutine calls/returns
  - a stack is a structure for organising data
    - we access a stack by pushing onto the top and pulling off from the top - First-In, Last-Out architecture
  - put the next function call on top of the stack