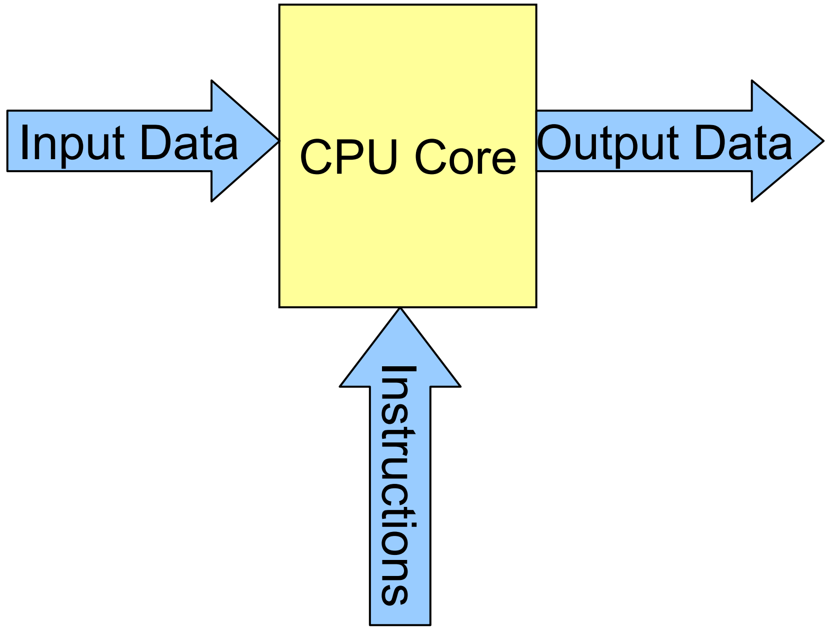
Embedded Systems Design

Processor Choice Lecture Notes

Processor Metrics

- The basic CPU model:

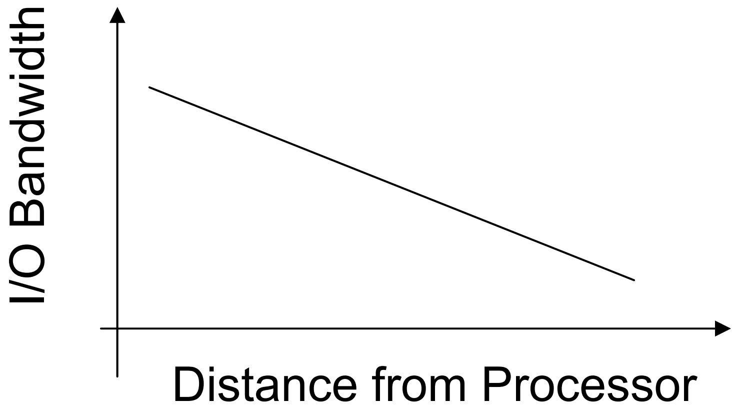


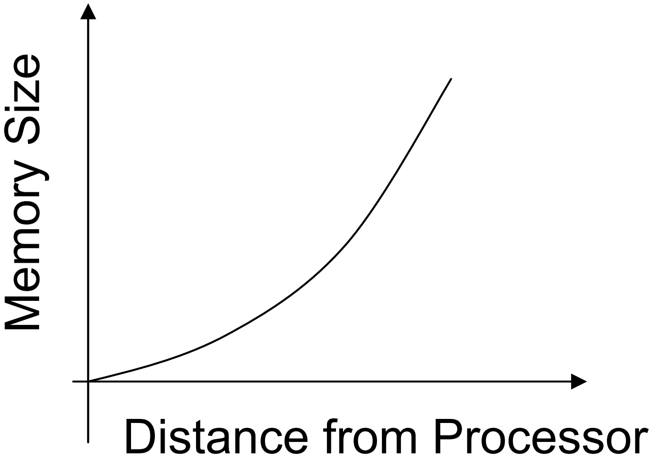
CPU Metrics

- The rate at which instructions are completed (instructions retired rate) depends on:
  - How fast the core can process an instruction
  - The availability of data and instructions to the core – a function of the CPU I/O subsystem

CPU Metrics

- Large, fast memories are difficult to make (low silicon yields) and is therefore more expensive
- The result is that small, wide and fast memories are placed closed to the processor and slower, larger and narrower memories are placed further away from the processor



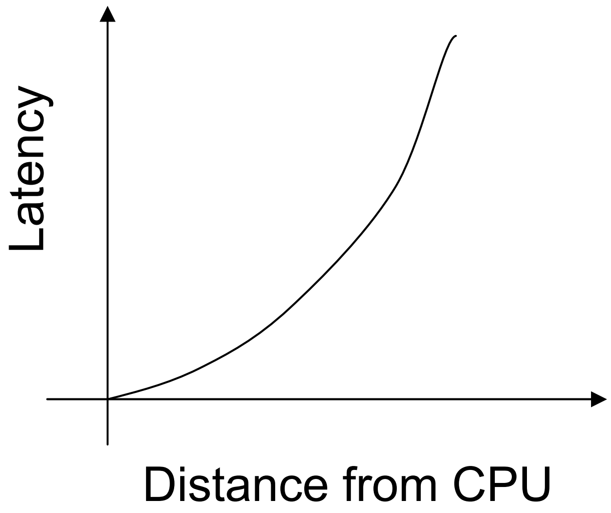


CPU Metrics

- Also running wide and fast signals over significant distances radically increases the cost of the PCB real estate.
  - I.e. An A4 sized PCB 2 layer PCB may be <R1k. The same size in 12 layers with exotic high density mfg technology is >R30k.

CPU Metrics

- Thus the further away from the CPU one goes the slower is the I/O and the larger the memories.
- Therefore the I/O substructure speed depends on where in the structure the data is sourced from / written to – the further away the more latency is introduced,:

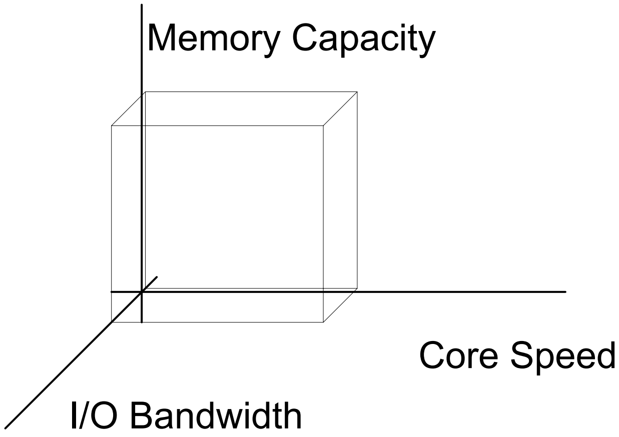


CPU Metrics

- An important exception to this rule is the capability that is provided on-board the CPU
  - I/O is relatively cheap
  - And so (within bounds) memory if included either on the silicon die, or as an Multi-Chip Module (MCM) within the CPU packaging

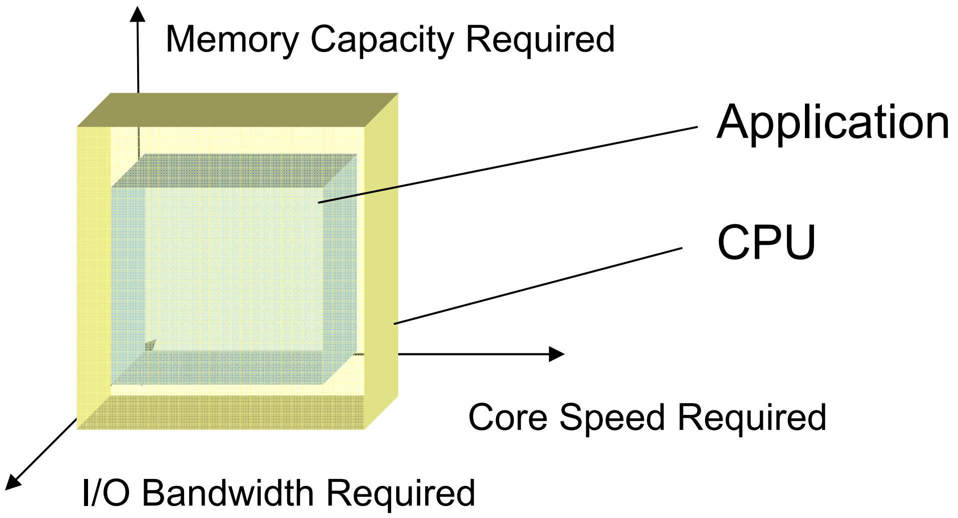
CPU Metrics

- It is however clear that the defining metrics of CPU performance are
  - Core speed
  - Memory size provided both on and off chip
  - I/O Bandwidth supported
  - Can be graphed as shown below – volume indicates the processor performance envelope



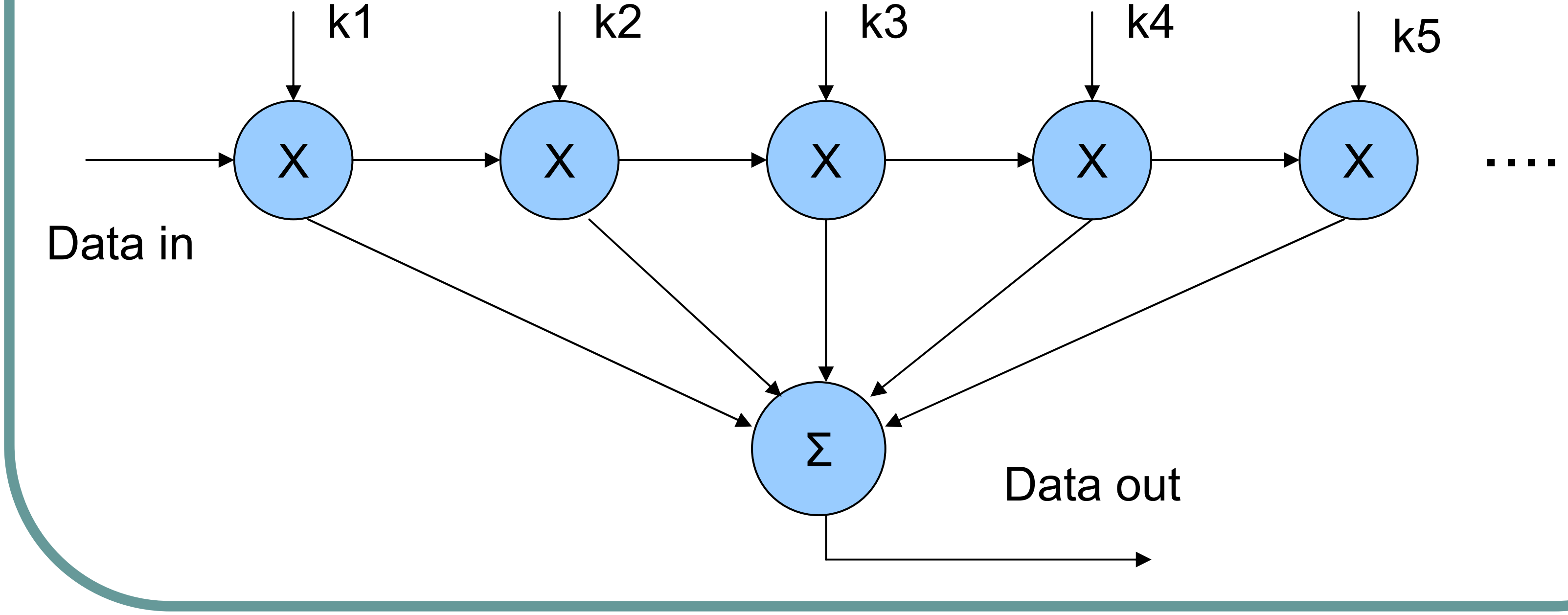
Application Metrics

- The application can now be characterised according to the same metrics
  - Core execution speed required
  - Memory capacity required
  - I/O Bandwidth required
- If the application volume fits within the CPU volume, the CPU should technically be sufficient for the task



An Example

- So let's do some calculations..
  - Take the example of a 64 tap FIR filter



An example

- The FIR filter samples 12 bit data at a 5 MHz rate => 5 MSPS
- In each cycle the application must:
  - Shift the delay line up one place (assume use of pointers, i.e. Store one data value to 16 bits and update one pointer -> 4 instructions)
  - Calculate 64 multiplies (16 bits each)
  - Add 64 values together
  - Store the result
- Thus the application requires
  - ~ 132 assembly instructions x 5 MSPS = 660 MIPS core instruction rate
  - Memory = 64 data values, 64 filter coefficients, 1 output value = 129 words or ~ 260 bytes, assuming the output need not be buffered to any length
  - I/O Bandwidth = (1 x 16 input, 1 x 16 output, 64 x 16 bit coefficients, 64 x 16 bit multiply outputs) x 5 MSPS = 130 x 16 bit x 5 M = 1.3 GB/s transfer rate required

An example

- The latter values are then the minimum requirements for a single CPU solution to the problem.
- High internal processing and I/O bandwidth is required – reason why FPGA’s do better at FIR filters than CPU’s