Computer Design Supervision 3

Matthew Johnson
Trinity Hall
mjj29@cam.ac.uk

January 27, 2003

1 y1998p4q6

1.1 busses

The memory bus on a typical PC is designed to be as close as possible to the processor. The latency needs to be minimal on the bas - particularly from the cache memory, and the bandwidth is also designed to be very high. Usually the bus runs at the same speed as the system's Front Side Bus.

Device interfaces are typically slower, but may but up to the FSB speed in high end machines. The devices connected will have very high latency anyway, so the bus latency is less important, but you will usually want to transfer large amounts of data at the same time, so high bandwidth is good. DMA devices access memory directly, so there will be connections from this bus to the memory bus, without needing CPU intervention.

I/O busses tend to have older, slower external devices connected to them. They normally connect via one of the device interfaces, and will run at a much slower clock frequency.

1.2 serial/parallel

Serial transmissions uses a single data line and transmit one bit at a time, where as parallel uses several data lines, and can transfer many bits at once.

1.3 USB

The USB uses a bi-directional differential signalling twisted data pair, compared to the signal serial data line of RS232.

1.4 Plug & Play

USB defines a standard for identifying peripherals by vender name & id - which RS232 does not.

2 y1998p6q2

2.1 memory heirachy

Memory is usually organised in a heirachy of memory types, that start fast and small, and get progressively larger and slower. Fast memory is usually very expensive and can't be bought in large quantities, but it can be used as a buffer for cheaper memory, improving access performance to it.

2.2 characteristics

Memory Type	Size	Latency
Register File	256bytes	1 cycle
L1 Cache	64kb	3 cycles
L2 Cache	256kb	11 cycles
Main Memory	1Gbyte	100 cycles
Hard disk	100Gbytes	20,000,000 cycles

	Memory Type	Size	Bandwidth	Latency
E.g:	Register File	256bytes	1.6GBytes/sec	10ns
	Main Memory	1Gbyte	1.6GBytes/sec	38.75ns

2.3 Direct Mapped Cache

A Direct Mapped Cache hashes the memory location of the block to a unique place in the cache. This may conflict with another block already in the cache. In that case the existing block in the cache would have to be flushed, and the new one inserted instead.

2.4 Set-Associative Cache

A Set-Associative cache hashes to an area of the cache, This will reduce cache conflicts, because there are several positions for a block to go in. In this case if there is still a conflict, then the cache can decide which block to flush. The least recently used, or the oldest block, might be sensible ones to be replaced.

3 y1999p4q6

3.1 Registers

General purpose registers allow a lot more flexibility in programming, Tying specific registers to specific instructions or programming concepts has problems when trying to nest said constructs or instructions, and reduces what an optimising compiler can do.

3.1.1 example code

Accumulator machine:

```
      load  #0
      store [#1]  ; a=0
      load  #1
      store [#2]  ; b=1
      load  #0
      store [#3]  ; i=0

loop:
      load  [#1]
      add   [#2]
      store [#1]  ; a=a+b
      load  [#1]
      sub   [#2]
      store [#2]  ; b=a-b
      load  [#3]
      add   #1
      store [#3]  ; i=i+1
      cmp   #5
      jmpne loop
```

Register machine:

```
      ldr   r0,#0    ;a=0
      ldr   r1,#1    ;b=1
      ldr   r2,#5    ;i=5

loop:
      add   r0,r0,r1  ;a=a+b
      sub   r1,r0,r1  ;b=a-b
      subs   r2,r2,#1 ;i=i-1
      bne   loop
```

4 y1999p6q2

4.1 RS232

The ASCII character “A” would be transmitted over RS232 by sending the start bit, then the 8 bits of the ASCII value of “A” (65), then a stop bit. If the default is the data line high, and the start bit is low, then then data would be sent as 0010000011 - the first and last bits being the start & stop bits.

4.2 DRAM

DRAM is very efficient at reading several consecutive memory locations, as it has a very wide memory bus, so that several can be read at once. The DRAM is broken into several banks so that it can read from consecutive words simultaneously, and send them on the bus. The CPU, or its memory controller, gets a request for a block of ram. This strobes the Row Address lines, and selects which row within the memory will be read. Consecutive blocks can be read by loading several values onto the Column Address line, without having to reload the row address.

4.3 cache

Cache is a small about of fast memory, typically on the processor die itself. Because it is on die it usually has orders of magnitude less latency, and large bandwidth. This makes it preferable that data should be read from there if you can not from RAM.

A lot of memory accesses in programs have spatial locality - in that they happen to contiguous memory locations. Hence, when a memory access is made, the cache will read in blocks near the requested location, because they are likely to come up as well.

5 y2000p3q6

5.1 Moore’s Law

Moore’s law is a principle suggested that processing power should double every 18 months, due to improvements in silicon manufacturing techniques. It has held approximately true for several decades.

5.2 Memory vs CPU

While CPU speeds have been increasing with Moore’s Law, memory access speeds haven’t. Processor manufacturers have made up for this using tricks such as SIMD and MIMD instructions - which operate on lots of data at once.

Increasing cache size and number of registers can also reduce the amount of data that needs to be transfered to or from main memory.

6 y2001p6q2

6.1 tlb

A *translation look-aside buffer* stores mappings between logical and physical pages. Since most programs only use a few pages of memory, or use a few pages regularly, the cache can have good performance with only a modest size.

7 y2002p5q2

7.1 cache types

A direct mapped cache makes a hash of the memory location with a unique cache location. Thus cache clashes must result in remove the earlier value from the cache. Associative mapping defines blocks of cache locations with the same hash value. Thus small clashes don't cause data to be removed from the cache, but each block must be searched sequentially to find the cached data.

7.2 burst mode caches

Caches exploit spatial locality of memory accesses to pre-fetch data from memory, by reading a large chunk of memory together in one burst. This is likely to work because memory accesses are often close together in RAM.