MEMORY: RAM & CACHE

CONTENTS:

- DRAM organization
- Read timing
- Latency
- DRAM types (DDR and variants)

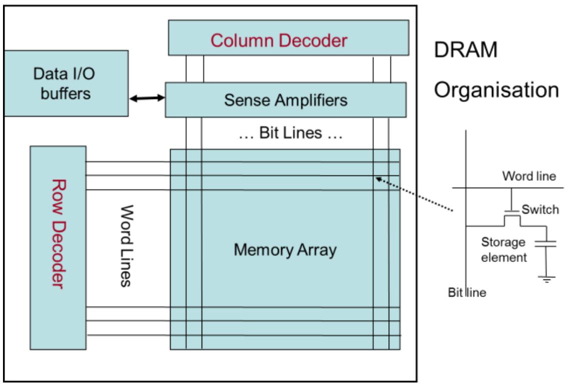
INTRODUCTION

The main memory in a PC consists of DRAM (Dynamic Ram). This is memory that has to be refreshed at regular intervals. A number of chips are typically assembled onto a module (e.g. a DIMM: Dual In-line Memory Module). 8 chips, each providing 8 bits, will give the 64 bits needed by a Pentium. A memory controller (which probably also talks to the graphics system) interfaces to the memory.

QUESTIONS

In the Apple II computer, the DRAM was clocked faster than the processor. Since then, DRAM clock rates have increased by about 9% per year, while processor clocks increased by about 50%. Why the difference?

What is the purpose of DRAM (where does its data go?)

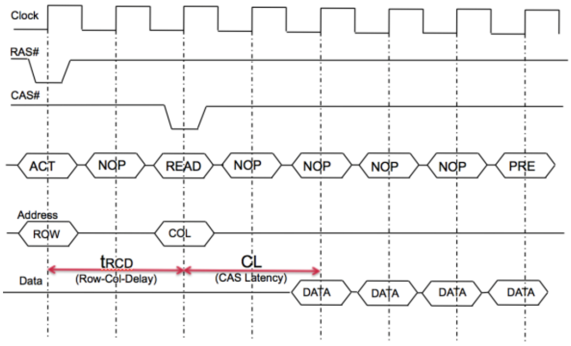


DRAM TYPES

The RAM in a PC is now a major bottleneck, working at a much slower rate than the processor. A lot of different types of RAM have been introduced to try and improve this situation. Cache is made from fast SRAM, but is too expensive for the PC's main memory, so DRAM is still used. It is organised so that the address is split into two halves: row and column.

ROW & COLUMN

A 256 M chip needs a 28 bit address. To save on pins, this is presented as two halves: a row, then a column address. Internally this is gated to the Row Decoder, then the Column Decoder. In a basic DRAM this cycle of Row, then Column address is repeated for every cycle.



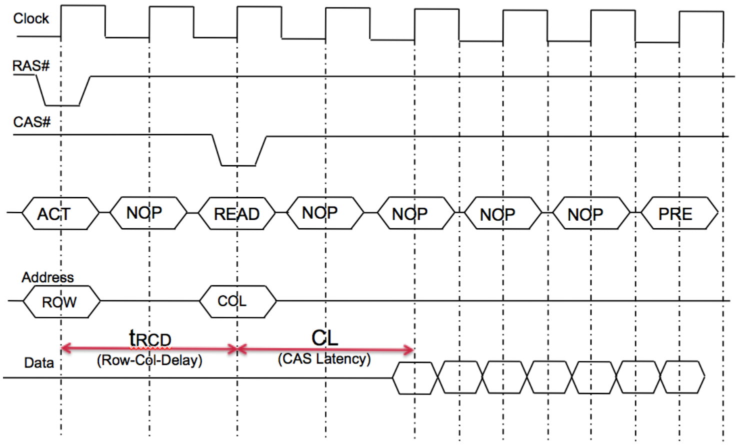
SDRAM

In early chips, the data came out of the RAM after a delay. In current chips, the data is read at a time synchronised to a clock signal, hence the name SDRAM: Synchronous DRAM. The SDRAM chip is normally classified by the clock rate it can operate at in MHz (e.g. 1600MHZ). A typical memory module might be a "CORSAIR DDR3 1600 PC Memory - 8 GB DIMM RAM" (£60 from PC World, July 2015). Once the first data has been found, consecutive data can be read out quickly without sending new addresses.

DDR: DOUBLE DATA RATE

Current versions of SDRAM can send data to the processor twice per clock cycle: 'Double Data Rate' or DDR SDRAM. Both edges of the clock cycle are used. So, 'DDR667' would be used with a 333 MHz basic clock. The naming scheme is different. Now it refers to the bandwidth: PC6400 means 6400 MB/sec (800 M transfers/sec * 8 bytes). You need to remember the 64 bit bus width, i.e. a maximum of 8 bytes per transfer.

DDR DRAM READ TIMING



LATENCY

It is not quite that case that the clock rate equals double the rate that data can be read out. There is always a latency caused by the need to provide the row, then column, addresses. This initial delay has not improved much recently (the best chips have gone from about 50nS twelve years ago to 10 nS just now). Having found the first block of data, subsequent data can be read out very quickly: that has improved by factor of more than 10 over the same period.

DDR2 & DDR3

DDR2 & DDR3 produce data at double the clock rate, just as with DDR memory. The modules have a different number of pins, can run at faster clock rates and use less power. DDR goes up to about DDR400, DDR2 starts at DDR2-400 and goes up to about DDR2-1300. DDR3 goes up to DDR2400.

CLOCK RATES/BANDWIDTH

The memory is now referred to, either by the DDR clock rate (DDR400), or by the maximum throughput, (PC3200). As 8 bytes are transferred at a time, the value is 8 times the 'DDR' clock rate. Note that, because of latency, the actual throughput will be different. Remember also that the 'DDR' clock rate is double the actual clock rate. For instance, DDR400 use a 200 MHz clock. As an example, a memory module could be classed as DDR2-800. This means the peak transfer rate is 800MBytes/sec, running from a 400 MHz clock. This might also be called PC2-6400 (8x the DDR figure, as 8 bytes are transferred at a time.

GDDR (& LAPTOPS)

Graphics cards often use specialised DRAM chips: GDDR2, GDDR3 or GDDR4. These are similar to DDR2 and DDR3. They are not identical. For instance, they aim to run faster, and may use a higher voltage to achieve this. This can lead to higher power consumption. Also they are generally more expensive.

Laptops tend use memory modules with a smaller form factor: SODIMM (Small Outline Dual Inline Memory Module).

CACHE MEMORIES

CONTENTS:

- Cache features
- Multiple levels
- Cache read and write strategies

INTRODUCTION TO CACHE

Cache memory sits between the processor and Ram. With current processors, RAM is too slow to work at the full clock speed of the processor. Many of the principles of cache memory are similar to those of virtual memory (principle of locality, replacement strategies). The key difference is that cache has to work in something like 5 nanoseconds. There is no time to run software; everything must be done in hardware. So, unlike virtual memory, the operating system is not involved.

Cache memory is relatively expensive SRAM. Typically the cache is one or two per cent of the size of the main RAM. The principle of locality ensures that this is sufficient for at least 80-90% of the memory accesses to go successfully to cache. The Principle of Locality states that memory accesses tend to be grouped together. If you have just fetched one line of code, you are likely to fetch the next line soon. If you have just fetched one pixel from an image to process, you are likely to want to fetch neighbouring pixels. For most of the time, memory accesses go to a small set of locations: these are likely to fit in cache.

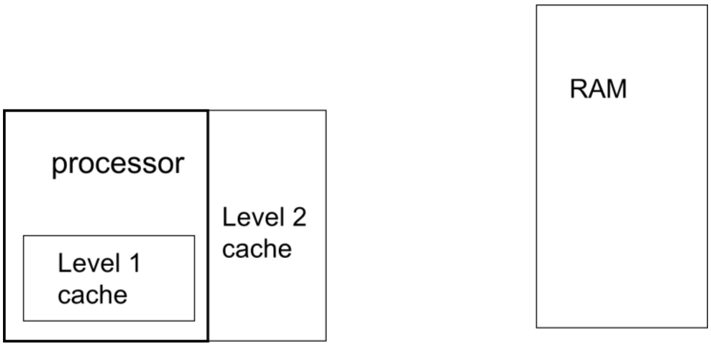
MULTI-LEVEL CACHE

On a high-end processor, there may be a small amount (perhaps 128 Kbytes) in the heart of the processor chip, with a larger amount (2-4 Mbytes or so) as a second-level cache. The second level cache will be on the same chip as the processor, but at the side, rather than in the heart of the processor. In early PCS, this cache used to be on a different chip but the delays of sending signals down copper tracks have become relatively too large.

How fast does an electrical signal get along a pcb track in 1 nS? (approx speed about 1/3rd of the speed of light).

SEARCHING CACHE

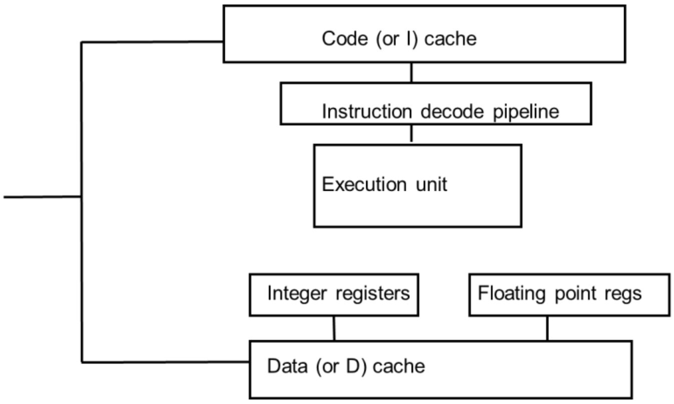
L1 Cache is inside the processor core; it is searched first
L2 cache is at the side of the processor; it is searched if the data can't be found in the L1 cache.



Level 1 Cache

The L1 cache built into a processor is usually split into separate sections for data and code; the processor can then fetch from both caches at the same time (Harvard architecture). This is possible because inside the processor, instructions and data are going to different sections of the processor. Second level cache is usually organised as one block that will store code and data.

Splitting L1 cache into separate Instruction and data cache can speed things up. Why is this not done with level 2 cache & RAM?



CACHE LINES

When data is fetched from cache, it is fetched in lines. Typically these are blocks of 16, 32 or 64 bytes. The processor may only request one instruction (perhaps three bytes), but actually gets perhaps the next 10 or so instructions. This works well with DRAMs, which are slow to produce the first data, but then quick (provided the data is consecutive).

CACHE WRITES

When data is written to the cache, there are several ways of handling the transfer to the main RAM. The simplest to implement is to "write-through" the data, i.e. write the data to the RAM every time the data is written to the cache. This means that the RAM is kept up-to-date with the cache (handy if there are other processors using the bus that might need to see the contents of RAM). The disadvantage of this is that the RAM may be written to more than is strictly necessary. For instance, there may be a whole sequence of changes to a variable or flag that the RAM doesn't need to know about if they are all happening to the variable/flag when it is in the cache.

The other technique, "write-back" or "copy-back", only writes data to RAM if the cache is being emptied to make room for new data. This is more complicated to implement, but gives better performance. It does lead to possible problems if there are other processors (e.g. a DMA controller) that use the memory. They will need to know if the contents of RAM are up-to-date. In practice, the problem is not too great, since there are always more reads than writes.

Question: why?

CACHE TAGS

Cache is organised into lines, typically containing 16, 32 or 64 bytes. A line is written to/read from memory as one unit. Since cache is smaller than main memory, only some of the contents of the main memory are present in cache at any one time. Tag bits are used to identify what section of main memory is present in cache at any time. There are actually several distinct ways of organising the cache.

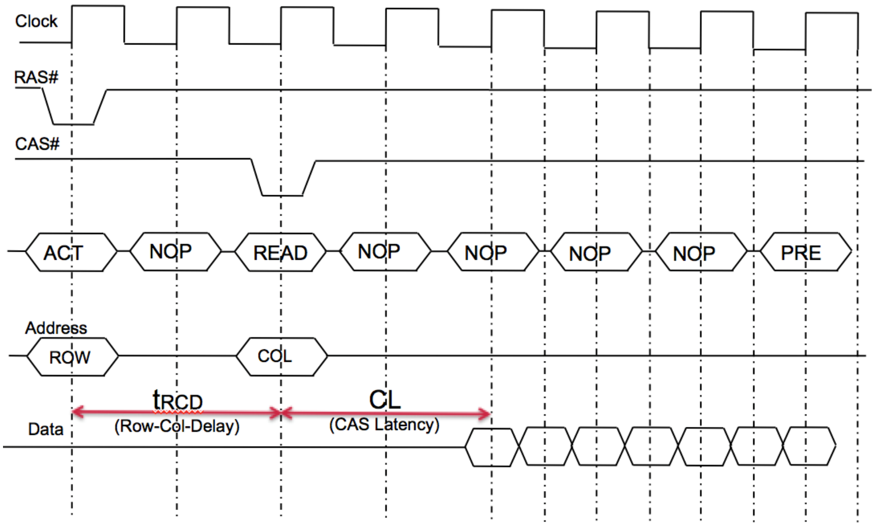
FURTHER READING: DRAM TIMINGS

DRAM timings are complicated. Generally 4 of the most important timings are given, though there are more times that you may come across. A standards body, JEDEC, has laid down how memory modules must work in order to be referred to as ‘JEDEC standard’. Typically four numbers are given, e.g. DDR3-800E memory might be referred to as 6-6-6-15. Each of these numbers gives the numbers of clock ticks for particular delays. The four figures are for t_{CL} – t_{RCD} – t_{RP} – t_{RAS}. These are abbreviations for:

- CAS latency or t_{CL} the delay between sending a column address and the data appearing.
- RAS to CAS delay or t_{RCD} the delay between the RAS & CAS signals
- Row Precharge time or t_{RP} How fast access can move from one row to the next
- RAS active time or t_{RAS} the time between a row being activated then deactivated

The clock ticks refer to the basic memory clock rate (not the double data rate that data is read from the DRAM). In the DDR3-800E memory referred to above, the actual clock rate would be 400MHz, i.e. 2.5 nS per clock tick. So, a t_{CL} of 6 means 6 x 2.5 = 15 nS

Here is what the most important of these look like on the earlier diagram:



The exact timing will depend on what is going on. A lot of the time we are at the correct row and column and the data comes out every half clock (1.25 nS in our example). We May be at the right row, but have only just specified the column and there will be a delay of t_{CL}. In the worst case we may shift to another row and column (a Random access) and we have to wait for t_{RCD} & t_{CL}. In the above example (6-6-6-15), this would be (6+6) x 2.5 = 30 nS.

On a memory module, the possible timing configurations are stored on a small permanent memory and read out via a few serial lines. This is done at boot time so the motherboard can determine what settings to use. The system is referred to as SPD (Serial Presence Detect).