Engineering Tripos Part IIA THIRD YEAR

Module 3F5: Computer and Network Systems

Solutions to 2013 Tripos Paper

Authors: Andrew Gee and Tim Wilkinson

1. **Parallel processing, false sharing in caches**

(a)

SIMD, Single Instruction Stream Multiple Data Streams. A computer classification in Flynn's taxonomy of parallel processing machines. Multiple execution units respond to the same instruction at the same time, but operate on different data. Useful for vector processing, commonly used in graphics hardware.

MIMD, Multiple Instruction Streams Multiple Data Streams. A computer classification in Flynn's taxonomy of parallel processing machines. Multiple uniprocessors connected on a single bus or via a network. The most general form of parallelism.

SMP, Symmetric Multiprocessor. A type of single address space multiprocessor in which accesses to main memory take the same amount of time no matter which processor requests the word and no matter which word is requested.

UMA, Uniform Memory Access. Means the same thing as SMP.

NUMA, Nonuniform Memory Access. A type of single address space multiprocessor in which some memory accesses are faster than others depending on which processor asks for which word.

SMT, Simultaneous Multithreading. An operating mode for superscalar hardware that allows several processes to run concurrently, with instructions from distinct processes fetched in the same clock cycle. Compared with conventional process switching, the superscalar hardware spends less time idling since there are fewer dependencies between the running instructions. [20%]

(b) A small number of processors can be connected together as in Machine A. Since each processor has its own cache, the single bus and memory system can serve the needs of all the processors, as long as there are not too many of them (up to a few tens of processors). With more processors, the single bus and memory system becomes a bottleneck, and the need for a physically longer bus also reduces the bus's bandwidth and increases its latency. Hence, architectures like Machine B tend to be used for larger MIMD machines. The memory is distributed amongst the nodes, so local processor-memory traffic can proceed at a high rate, independent of the number of processors. Inter-processor communication, however, is over a network and slower than in a single bus design. [20%]

(c) Write-invalidate is usually preferred to write-update for bus efficiency reasons. Multiple writes to the same cache block with no intervening reads require multiple update

broadcasts in an update protocol, which is wasteful of the limited bus bandwidth. In contrast, the invalidate protocol requires only a single invalidate broadcast on the first write. Minimizing bus traffic is of paramount importance in this sort of architecture, since the bus is usually the bottleneck, limiting the number of processors that can be accommodated. [20%]

(d) The code is highly susceptible to false sharing of the `partial_sum` array. This is a small array and most likely resides in a single cache block. Elements of `partial_sum` are read and then written 2,500,000 times by each of the four threads. Every write will cause the cached copies in the other cores to be invalidated. The next core to attempt a read will therefore get a cache miss, and will need to read the block from main memory, which must first be updated from the most-recently-written-to cache. Essentially, every increment of an element of `partial_sum` will cause (i) a write back of a cache block to main memory and (ii) and fetch of a cache block from main memory, so two memory transactions. It would be as if the caches were not there.

Not only do we lose the benefit of the caches altogether, but memory accesses might stall given that we have four processors simultaneously thrashing the memory system. There is also the overhead of instantiating the four threads. Indeed, it is not impossible that the parallel code might run *slower* than a simple, single-threaded implementation.

The fix is to accumulate the partial sums not in a shared global array, but in a simple variable local to each thread. Such local variables will not reside in the same cache block and will therefore not result in false sharing. Even if the programmer does not get this right, a decent compiler might fix things up anyhow, but only if compile-time optimizations are enabled. [40%]

2. **Virtual memory systems, page tables**

(a) There are two main requirements that motivate the adoption of a virtual memory system. The first is the desire to be able to write programs without having to worry about the amount of physical memory installed in the computer. The second is the need for the CPU to execute multiple processes separately: each process should be unaware of, and protected from, the others. [15%]

(b) The need to look up address translations in the page table (which is large and therefore stored in main memory) undermines the existence of the cache. So the CPU typically contains a small, fast cache called the translation lookaside buffer (TLB) which caches recently used page table entries. Locality of reference to the page table means that the TLB miss rate is very low, typically 0.01%–1%. The TLB block size might be 1–2 page table entries, its hit time is less than 1 clock cycle, and its total size might be 32–4096 page table entries. [15%]

(c) The page size is 4 KB $= 2^{12}$ bytes. There are a total of $2^{64}/2^{12} = 2^{52}$ virtual pages. The number of physical pages is much smaller: 512 MB $= 2^{29}$ bytes, so $2^{29}/2^{12} = 2^{17}$ physical pages. Each page table entry contains a physical page number (17 bits) along with valid and dirty bits (2 bits). Rounding up to the nearest whole byte, each page table entry is 3 bytes long. So the total size of a page table is $2^{52} \times 3 = 12288$ TB. Each process needs

its own page table, giving 1228800 TB in total. Clearly, simple page table structures are impractical when the virtual address space is large. [20%]

(d) (i) By including a process identifier in the inverted page table entries, we can make do with a single, global page table shared between all processes. The inverted page table would then contain all the information we need: one entry per physical page, showing which (if any) virtual page is mapped to it. The process identifier is necessary since different processes might use the same virtual addresses. [15%]

(ii) Each entry holds a virtual page number (52 bits), a process identifier (16 bits), and valid and dirty bits (2 bits). Rounding up to the nearest whole byte, this gives 9 bytes per entry. There are 2^{17} entries (one per physical page), giving a total memory requirement of 1.125 MB. [15%]

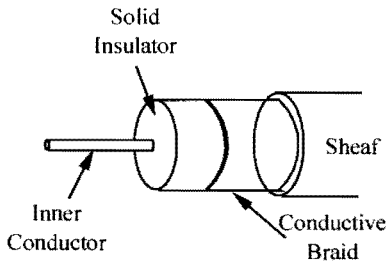
(iii) The traditional arrangement allows rapid address translation: the virtual page number points to a unique row in the page table, from which the physical page number can be retrieved (if valid): so just one memory access. In contrast, the inverted page table is not indexed by the virtual page number. Naively, we could perform a linear search of the entire table, checking every stored process identifier/virtual page number to see if it matches the one we are trying to translate. If we get to the end of the page table without finding it, we have a page fault. So that's 128K memory accesses to identify a page fault, and on average 64K memory accesses for a successful translation.

With more sophistication, we could employ some sort of hashing strategy to decide where in the inverted page table an entry should be stored. [20%]

Elaboration, not expected of candidates. The most common approach uses a hash function to map process identifiers and virtual page numbers to indices into a *hash anchor table*. Each hash anchor table entry contains an index into the inverted page table. Ideally, then, each translation requires the calculation of a hash function and then just two memory accesses (the first to access the hash anchor table, the second to access the indicated row of the inverted page table). Inevitable collisions mean that the required translation might not be stored at the indicated row, but nearby, so in practice we might require, on average, 2.5 memory accesses per translation. Note how the hash anchor table allows a a virtual page to map to any physical page, which is essential to avoid unnecessary page faults. This would not be possible if we hashed directly into the inverted page table.

Question 3

a) A coaxial cable consists of a stiff copper wire as an inner conductor, inside a solid insulator, that is itself inside a closely woven braided wire mesh that acts as an outer conductor. This is then covered in an insulating protective cover.



It is possible to use Maxwell's equations to calculate the electric and magnetic fields within the coaxial cable to give the attenuation in the cable due to the inner conductor and the solid insulator, along with the frequency cut-off for a given cable. The solution of Maxwell's equations also shows that coaxial cables have: High frequency cut-off. High immunity to interference and crosstalk.

Coaxial cable is a good transmission medium for high data rates over relatively short distances, however it is rather expensive and is not ideally suited to long distance transmission and has been largely replaced by optical fibre. It is now being used as the basis of most 'cable-TV' (CATV) systems as it has a high bandwidth and is much easier to join and terminate that optical fibres. It is extensively used in Ethernet, bus type systems where many nodes can be tapped into a single coaxial backbone. The size and quality of the coax dictates its base 10BaseXX. There are two main types of ethernet used with coaxial cable that are common place in local area networks (LANs). 10 Base 5 - data rate 10Mbits/sec over 500m, 10 Base 2 - data rate 10Mbits/sec over 200m.

The main limitations of coax are the cost and loss per unit length. In order to run at higher data rates, the quality of the coax must go up and so does the cost and loss. It also becomes more rigid and difficult to install. Twisted pair cables are cheaper and lower loss, especially when considering multiple pairs available per cable.

b) The two main areas dominated by twisted coax cable technology were early wide area networks such as X.25 and local area networks such as ethernet.

X.25 was one of the first data protocols to be well defined (OSI compliant), hence it forms the basis from which many later transport protocols have been developed. A key feature was the frame check sequence (FCS) as old co-axial cable lines were very poor quality and very prone to errors. The FCS field is a string of bits (added as a footer) which help determine at the receiving end whether the data in the frame has in any way been corrupted. Today's digital transmission technology is several orders of magnitude better quality than that of the 1970s, so that the heavy duty error detection and correction techniques used by X.25 have become redundant. Windowing and acknowledgement are now largely superfluous as the transmission technology of coax cable has been replaced by higher bandwidth and lower loss technology such as optical fibre. The replacement of coax at the physical layer prompted the transition from X.25 to Frame Relay. Instead of it being undertaken by the network, the job of error control or recovery is left to higher layer protocols. The frame relay network can then concentrate on raw data transport. The legacy of this process are the overall frame format and the use of the FCS still used today.

Flag	Address field	Control	Information field	Frame check sequence

The evolved frame format consists of five basic information fields specified by the data link layer.

Coax also was a dominant technology within LANs, through the evolution of Ethernet which originally used coax a shared bus mechanism in the physical layer. On a CSMA/CD (ethernet) LAN the terminals do not request permission from a central controller before transmitting data onto the transmission channel; they contend for its use. Before transmitting a packet of data, a sending terminal 'listens' to check whether a path is in use and if so, it waits before transmitting its data. Even when it starts to send data, it needs to continue checking the path to make sure that no other stations have started sending data at the same time. If the sending terminal's output does not match that which it is simultaneously monitoring on the transmission path, it knows there has been a collision. As a result of this collision process and the length of the coax cable, there is a limit on the length of a packet that can be transmitted before the packet will collide with its self as a result of a reflection from the transmission line termination of the coax cable. At 10Mbit/s using 10Base2 cable of 50m length, this was determined to be 1500bytes. Hence the MTU of Ethernet was set at 1500 bytes.

Ethernet has since gone on to dominate the whole LAN and now MAN and WAN systems and along with it has come the legacy of the 1500byte MTU. This can be seen in the sharp cut-offs seen in packet statistics even today when looking at modern network traffic.

This sort of limitation is not ideal under the OSI reference model for protocol design. All layers of the model should be defined as independent of one another and there should be no restrictions placed on higher layers based on lower layer performance. This is even more important when the limit is purely due to a legacy of the older evolutionary protocol design.

c) The bandwidth requirements of both the 10G and 100G Ethernet standards has meant that twisted pair cabling is not able to transmit data more than a few metres reliably. Optical fibre technology has the bandwidth capability, however there are still limits as to whether single or multimode fibres are used. Singlemode is faster but more expensive and more difficult to implement at both Rx and Tx ends simply and cheaply. MMF is more suited in terms of cost of both Tx and Rx integration with low cost electronics, however the demands of 100G Ethernet are still very challenging. Optical fibre will always have the upper hand over longer distances where loss is more important than cost.

Coaxial cable has come back as a possible solution to short distance 10G and 100G transmission up to the point where loss becomes an issue (~50m). A couple of issues that have been overcome by this adoption of coaxial cable are as follows.

Cost: a cheaper more flexible form of the cable has been developed which is capable of acting as a very reliable data transmission system up to 100G over short distances. This also includes clever encoding and driving schemes to maximise transmission

Performance: A key limitation was the transmission line action of the coaxial cable (especially reflections). This has been eliminated by using two cables as a full duplex pair. This also allowed a cheaper form of cable to be used.

Question 4

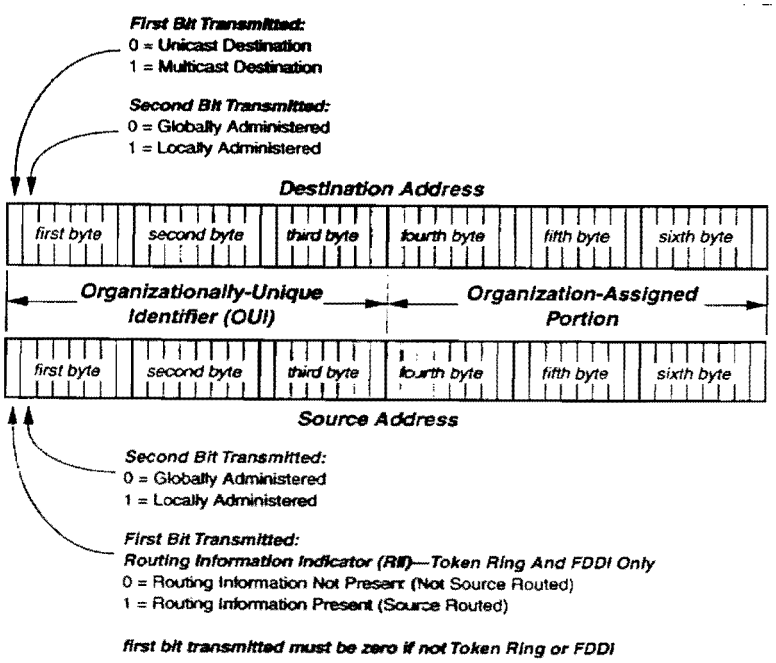
a) Routers are intelligent OSI layer 3 devices. They are designed to learn the topology of complex networks (even ones that are constantly growing and changing) and route frames or packet across them to the destination indicated in the header. Routers learn about network changes through experience. They each have a router table that contains the routes needed to connect addresses. These tables are regularly updated by experience and downloads from neighbours and major router table sites.

There are two main classes of router algorithms. Nonadaptive or static routing is where routes are learned from other routers on the network or download sites. This is a more efficient routing method as the overhead on the node is minimal, however there is more pressure on the node when an unknown address is received. For fixed routes, QOS is easier, but much less efficient if there are any changes or loss of routes.

Adaptive routing is where the contents of the router table is based on optimising traffic across the network. In this case, each route is selected based on network loads, topology changes and traffic densities. Optimisation often occurs with metrics such as number of hops, geometric distance or transmission time. If a router receives a packet or message for a destination which they do not recognise, they make a 'best guess' or choose a route at random and see if it is successful. They may also choose to flood the network by sending packets by all routes (except the delivery route).

The problem with networks of multiple routers (including the Internet itself) is that the individual routes through the network are difficult to monitor and manage. It is difficult to know which networks are being transited along the way, so that optimal network loading and security of information cannot be guaranteed. Also QOS is difficult as the overhead of processing routes is high and takes valuable time to implement. However if the route is lost or changes, then recovery is much faster and a lot less disruptive.

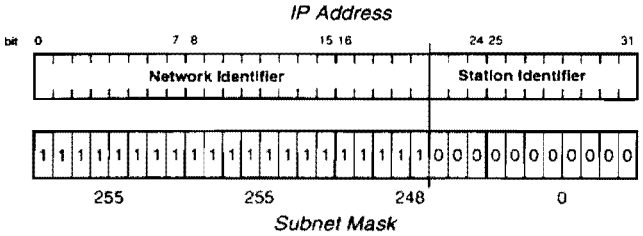
b) The key to a layer 2 MAC protocol is a uniform addressing structure so that LAN hardware can easily identify stations on the network and transmit packets between them.



The 48-bit MAC address is divided into two parts. The first 24 bits constitute the organisationally unique identifier (OUI), which indicates which organisation (typically the manufacturer) is responsible for assigning the remaining 24 bits of the address. In protocols such as Ethernet, the MAC address was originally designed to identify individual pieces of hardware on the LAN, however there are many issues when considering the scalability of a network using purely MAC addresses. There is no simple way of partitioning the addresses to differentiate between the network nodes and the stations. Because of this, to route MAC addresses, the routing table must contain all possible

addresses and their associated routes within the network. For a global network such as the internet this is clearly not scalable even with modern hardware and storage.

At layer 3 in a protocol such as IP, the source and destination addresses are both 32 bits. It is normal to write these addresses as four octets separated by dots (eg. 169.129.24.88). Each of the four decimal values may only have a value between 0 and 255. There are five old classes of IP address. Class A: Class B, Class C: Class D: Class E, for both unicast and multicast usage.



An IP address contains fixed-length fields which comprise of two main portions: The network identifier, which indicates the network on which the addressed station resides. The station identifier, which denotes the individual station within the network to which the address refers. IP station identifiers are locally unique, only being meaningful in the context of the identified network. Each IP address has an associated subnet mask of the same length (32 bits). The network identifier portion of the address is defined by the portion of the subnet mask set to 1's. Hence the layer 3 address can partition the network nodes away from the individual stations. This allows the routing process to use the coarse resolution network identifier to locate a network and then the station identifier to find individual stations. This is much more scalable in terms of a network such as the internet even on a global scale. The problem is the severe limited number of addresses within the 32bit space which limits the number of individual stations on the network. This has been rectified in 3 ways. 1) The IP address can be combined with the MAC address (ARP) 2) IPV6 has more addresses (64bit), 3) DHCP allows addresses to be recycled.

c) The two main features were the ability of the network to reconfigure itself and the rapid global adoption of the IP network address structure. The internet emerged from a US government and military initiative to enable the interconnection of different, mainly UNIX-based computer systems for intercommunication. The origin of the internet can be traced to the mid-60s with the creation of the US military network ARPANET. This was designed to produce a network that was capable of operating after a nuclear strike, in other words, a network that was capable of distributing the routing of data and not rely on a single vulnerable controlling station that could be easily targeted. The structure of the network was also to incorporate the then radical concept of packet switching to transmit the data from node to node. There was also a strong desire to utilise existing transmission media such as network structures and telephones of existing operators. The idea was to piggy-back the network as much as possible. A worldwide community of inter-linked computers quickly emerged. In 1992 the millionth host was added and by 1995 there were multiple backbones, millions of hosts and tens of millions of users. The size approximately doubles every year.

Being a globally unique address, the IP address allowed an end network and user to be identified, no matter how many transit servers, routers or networks would have to be traversed along the way. Along with the IP address came the domain name which identifies each unique address. The main drawback on scalability was the early choice of a 32 bit address which limits the modern scalability of the internet. IPV6 tries to alleviate this by going to a 64bit address, however it also has an effect on the

efficiency of the protocol. The main limit is the back compatibility which must be maintained with older IPV4 systems. This along with the need to be compatible with other IPV6 equipment manufacturers makes it very difficult to maintain a reliable internet QOS.

The main drawback of this adaptable structure is the lack of knowledge as to how the data is being transmitted and therefore the lack of control that one has when it goes wrong. The traffic flows in the network as a whole are therefore largely unmanaged and unmanageable. The only solution to slow response or congestion can be to add more capacity. Whether the capacity is added at the most appropriate point is a matter of chance. If a QOS is being maintained, then full control of the routing process is required, especially when things fail. This is very difficult with the internet as its tends to resist attempts at control by adapting against it. The only way that control is achievable across the whole network is if the equipment and interpretation of the protocols are common and/or proprietary to one manufacturer.