ENGINEERING TRIPOS PART IIA 2004

Solutions to Module 3F5
Computer and Network Systems
Principal Assessor: Dr F M Stajano
Second Assessor: Dr T Wilkinson

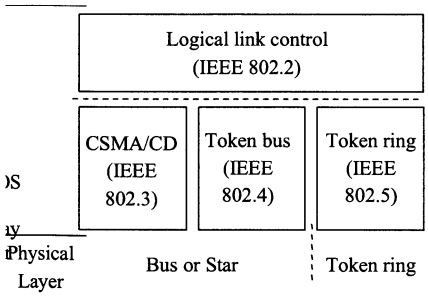
1 (a) Packet switch – statistical multiplexer of data. Directs data packets through network through either external control or by reading header. Used anywhere packet delay is not a constraint.

Protocol - for successful data communication across a network, appropriate operating procedures must be established. These procedures are called protocols. Any successful communication across a network must have a protocol.

Modem – a physical device which allows a higher data rate to be transmitted down a data channel than that set by the channel bandwidth. Eg V33 modem or ADSL.

Latency – any delay a packet incurs in transmission. Often incur latency when packets are buffered in a switch.

(b) CSMA/CD (IEEE 802.3, ISO 8802.3): Ethernet. The carrier sense multiple

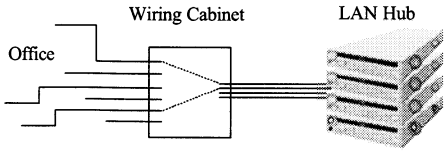


access with collision detection (CSMA/CD) protocol is a contention protocol. On a CSMA/CD LAN the terminals do not request permission from a central controller before transmitting data onto the transmission channel; they contend for its use. Before transmitting a packet of data, a sending terminal 'listens' 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. To receive data, the medium access control (MAC) software in each terminal monitors the transmission path, decoding the destination address of each packet passing through to find out whether it is the intended destination. If it is, the data is decoded, if not the data is ignored.

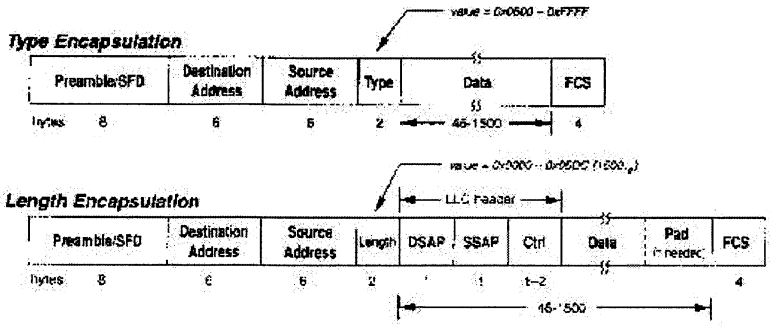
The use of a bus and CSMA/CD allows flexibility in the layout and number of users in the LAN. Theory suggests that random collision of a large number of devices can lead to transmission degradation under heavy traffic, however traffic is rarely random as most LAN transmissions involve a central server system. Traffic problems can be alleviated by subdivision into smaller LANs.

(c) The evolution of Ethernet to higher data rates has made CSMA/CD rather inefficient, however modern silicon VLSI means that a hub can now have intelligence and offer a port for each user and run each port at wire speed. Hence Ethernet MAC has become purely a frame format with no need for



access control. LAN hubs (or switches) have become necessary as the data rate is too high for shared media. This caused the development of LANs with structured cabling, using hubs and twisted pair (10 and 100baseT), in a star configuration.

When installed as part of a structured cabling scheme (nowadays the most common realisation of ethernet), twisted pair or coaxial cabling provides for the transmission medium. Multiple twisted pair cables are usually installed in each individual office and near each desk, and are wired back to a wiring cabinet. Next to the wiring cabinet is a LAN hub which replaces the coaxial backbone, so the arrangement is often called a collapsed backbone. The bus topology still exists, but only within the hub itself. If new devices are added, then they are patched through the wiring cabinet.
An Ethernet MAC sub layer frame can take two possible forms. The preamble/SFD, address and frame check sequence (FCS) are common to



both types. They are referred to as type and length encapsulation frame formats.

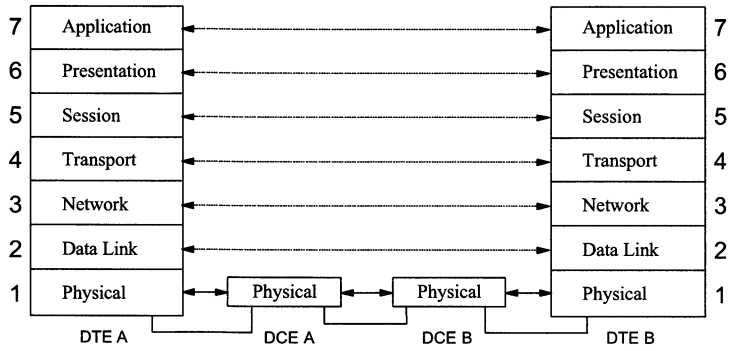
The key to MAC layer protocols is a uniform addressing structure so that LAN hardware can easily identify stations on the network and transmit packets between them. The MAC address has become globally standardised across the world. The simplification of Ethernet MAC along with the large number of users and the address format in each LAN has meant that the protocol is now used beyond the LAN into both MANs an even WANs. Once established it is very difficult to change a protocol, especially if it works well.

The explosion in the number of local area networks (LANs) and the need to interconnect LANs, as well as the growing number of client/server computing environments lead to the evolution of frame relay as a simplification of X.25. Frame relay provides for relatively cheap wide area data communication.

Flag	Address field	Control	Information field	Frame check sequence

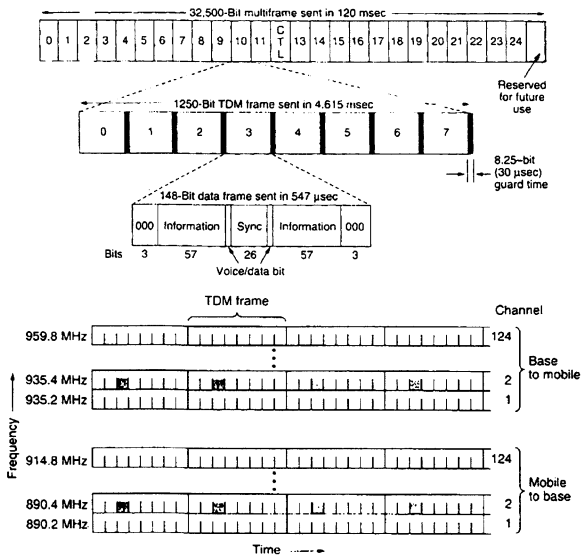
The frame relay frame format consists of five basic information fields, much like the data link layer format of X.25. The flag marks the beginning of the frame, delineating it from the previous frame. The address field carries the data link connection identifier (DLCI), the equivalent of the logical channel number (LCN) (ie is an OSI layer 2 address). Hence the structure of the frame relay is very similar to that of Ethernet, with a longer max frame length. A simple packet translation with fragmentation is all that is needed to allow Ethernet to be compatible with frame relay which means that Ethernet can use existing MAN and WAN equipment.

2 (a) The function of the physical layer is to provide a 'virtual bit-pipe' service to the data link layer of the OSI reference model. At the lower interface of the physical layer, we have the physical communications channel. This may be using any of the physical media available for transmission of data. Whatever the transmission media, the physical layer has to shield the data link layer from it and must adapt the output from the data link layer to a form that best fits the transmission medium.



When working within the lower levels of the OSI model, it is often useful to add a common communications device such as a modem to utilise the available communications channel. It is useful to split the lower end of the OSI model into two parts: The DTE and DCE

(b) Global Systems for Mobile Communications (GSM) was deployed before any of the competing American systems. It was simplest to make them pure digital, operating in a new frequency band (1.8 GHz). GSM uses both FDM and TDM. The available spectrum is broken up into fifty 200 kHz bands; within each band TDM is used to multiplex multiple users. GSM was originally designed for use in the 900 MHz band, but later frequencies were allocated at 1.8 GHz, and a second system closely patterned on GSM was set up, known as DCS 1800. GSM has up to 200 full duplex channels (200 kHz wide) per cell; each channel has a down-link and an up-link frequency.



Each of the 124 frequency channels supports eight separate connections using time division multiplexing (TDM). Each currently active station (mobile phone) is assigned one time-slot on one channel, theoretically 992 can be supported in each cell, but many of them are not available to avoid frequency conflicts with neighbouring cells. A single user can receive up to 9,600 bps of data.

All in all GSM is a fairly complex system; it handles channel access using a combination of TDM, FDM and a collision-based protocol known as ALOHA (an early form of Ethernet).

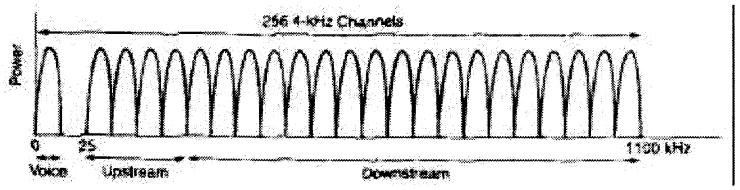
GSM is almost a pure physical layer process, however the conversion of phone numbers (addresses) is a higher layer process.

(c) If a V33 modem was used to send data, the data rate and error rate would be very bad. This s due to the high error rate of the physical channel in GSM. It is designed for voice transmission which is tolerant to errors, but if the channel is using a V33 modem, then the data will be corrupted in a very non causal manner. This means that a normal modem is not suitable:

Solution:
1) Employ extra error correction in both the modem data format and in the GSM channel system to remove and detect bit errors. This would increase error robustness but lower data rate. Overall would be better than pure V33.

2) Use a dedicated digital mobile data format such as GPRS and use as many available GSM channels as can be allocated. This improves both data and error rate, but is expensive and limits the number of high data rate users.

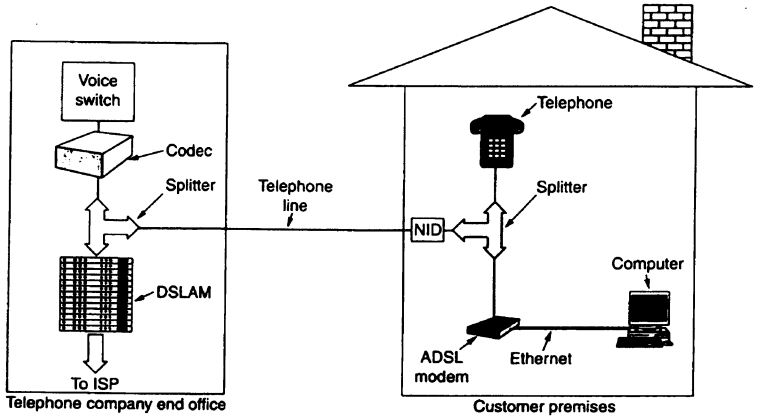
(d) The main limitation on the bandwidth of a Cat 3 UTP telephone line is filter placed at the exchange end before the voice signal is digitised. This limitation means that a normal modem must squeeze all of the data through a 3.7 kHz wide band. The Cat 3 UTP is in fact capable of transmitting data over short distances at frequencies in excess of 1.1 MHz, which is a property exploited by digital subscriber line modems.
A further enhancement on modem technology is the digital subscriber line (DSL) which allows the transmission digital data at a very high rate over



standard twisted pair cables up to a distance of around 5 km. There are a variety of standards such as VDSL, HDSL and xDSL, but one of the more common is the asymmetric digital subscriber line (ADSL) which typically offers a 512 bit/sec downstream and 64 kbit/sec upstream or 1 Mbit/sec downstream, 256 kbits/sec upstream for premium services.

The transmission coding is very complex and often adaptive. In the discrete multi-tone (DMT) system the basic 1.1 MHz bandwidth of the line is split into 256 4 kHz wide channels. The first is for voice services, followed by a gap of 5 before the next 250 which are used for data. A modulation scheme similar to V.34 (4000 baud) is used in each channel and each channel is monitored and the bitrate adapted to suite optimise the performance and compensate for and peculiarities in the line. With a high quality line a bitrate of 8 Mb/sec downstream is often possible.
ADSL is not suitable for mobile and wireless channels, as the dynamics of the channel bandwidth is very difficult to characterise and is difficult to split into smaller channels. xDSL is very similar to WDM in optical signal and relies heavily on a stable channel without reflections or multipath problems like those in mobile systems.

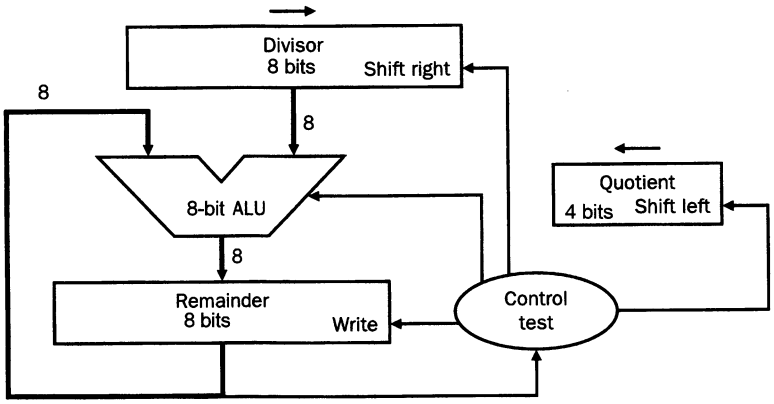
(e) A typical ADSL system is shown above with a network interface device (NID) set at the boundary before a splitter filters off the voice services. The

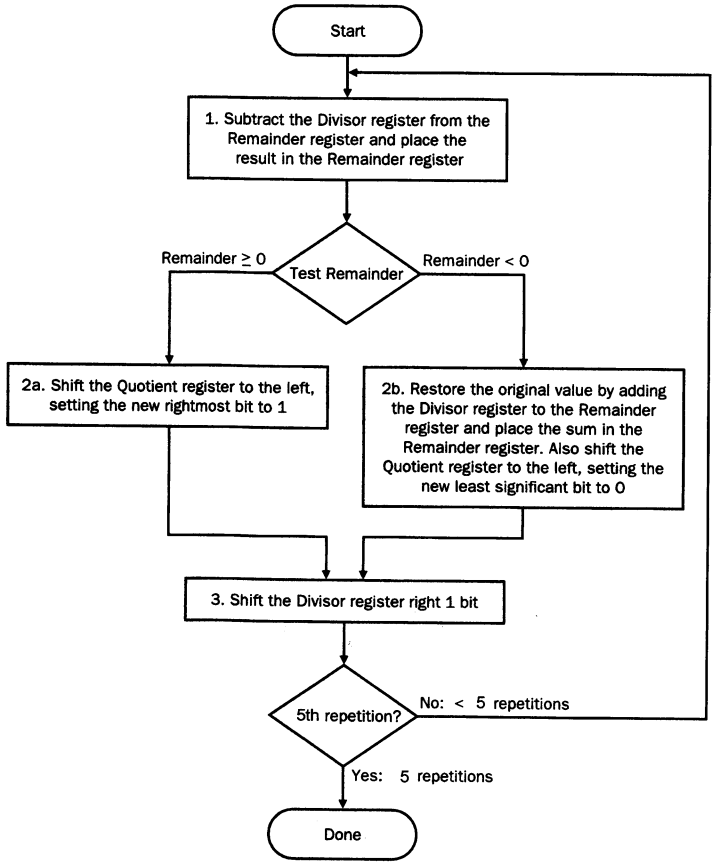


cable modem then connects to the computer via a standard such as Ethernet or possibly USB. A similar process occurs at the exchange end of the line with the voice channel being spit off and sent down the SDH or SONET. The other signals are passed through a digital subscriber line access multiplexer (DSLAM) which combines the channels and formulates packets for the data to be transmitted.

The two systems cannot be simply combined in the same network as there is the classic disparity of voice versus data services which means that no one network can reliably provide both acceptable latency and error performance over long distances.

3 (a) The divisor register, ALU and remainder register are double-width (8 bits), with only the quotient register being 4 bits wide. The 4-bit wide divisor starts in the left half of the divisor register and is shifted right 1 bit on each step. The remainder is initialized with the dividend. Control decides when to shift the divisor and quotient registers and when to write the new value into the remainder register.



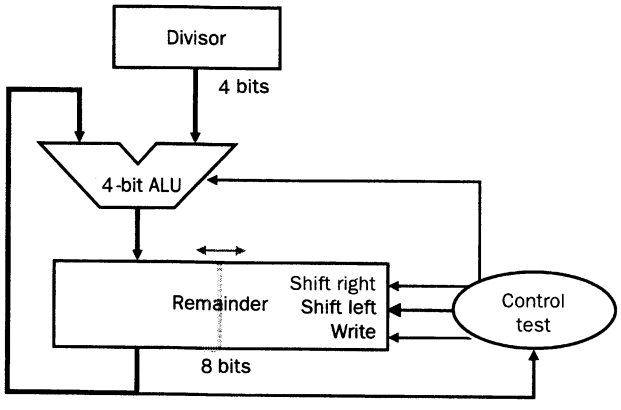


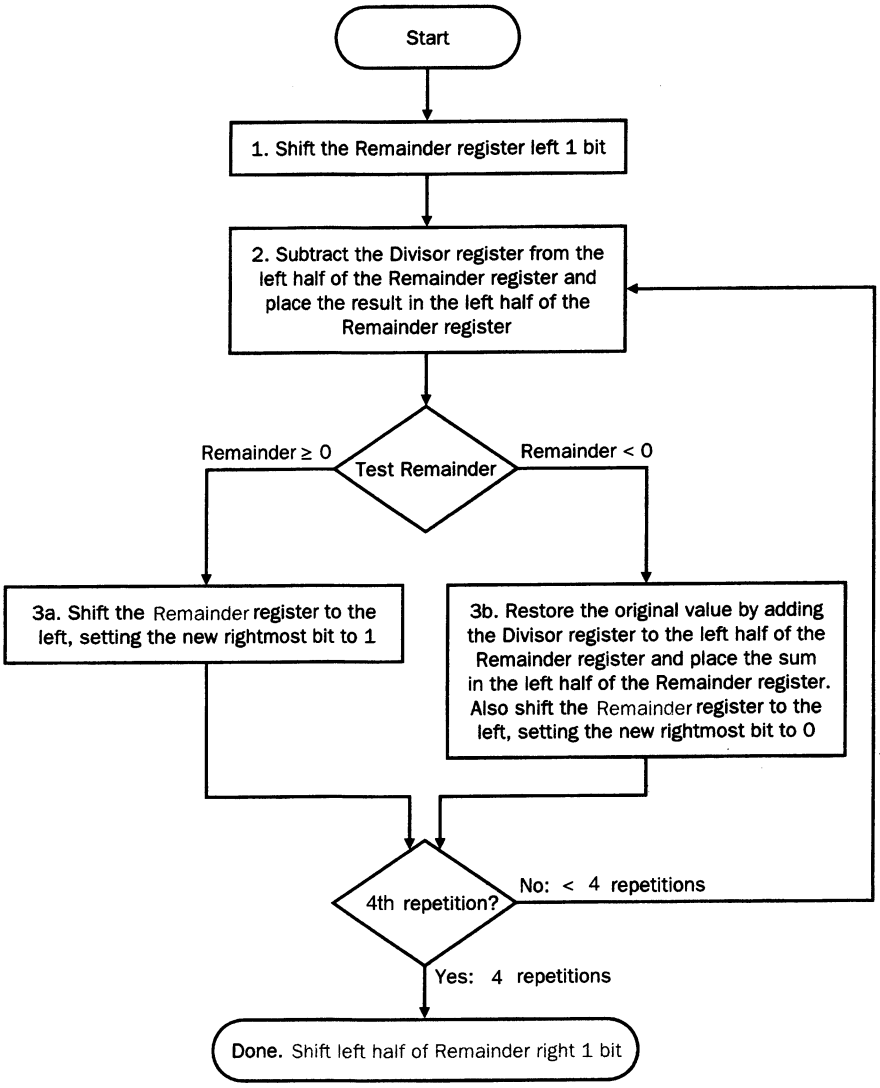
(b) Follow this example, which does it for seven divided by two.

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem – Div	0000	0010 0000	1110 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem – Div	0000	0001 0000	1111 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111
3	1: Rem = Rem – Div	0000	0000 1000	1111 1111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111
4	1: Rem = Rem – Div	0000	0000 0100	0000 0011
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011
5	1: Rem = Rem – Div	0001	0000 0010	0000 0001
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0011	0000 0010	0000 0001
	3: Shift Div right	0011	0000 0001	0000 0001

(c) The most important optimization is the one that allows us to halve the size of the ALU. This is achieved by shifting the remainder to the left instead of shifting the divisor to the right. Since the divisor no longer moves, it now fits in a 4 bit register which is one of the inputs to the ALU. The other input is the top half of the 8 bit remainder register.

Another possible optimization is to eliminate the quotient register. Since the 8 bit remainder register shifts left by one bit at each iteration, and its right portion is empty at the end of the division, the bits of the quotient (of which one is generated at each iteration) can be shifted into this register instead of a separate one.



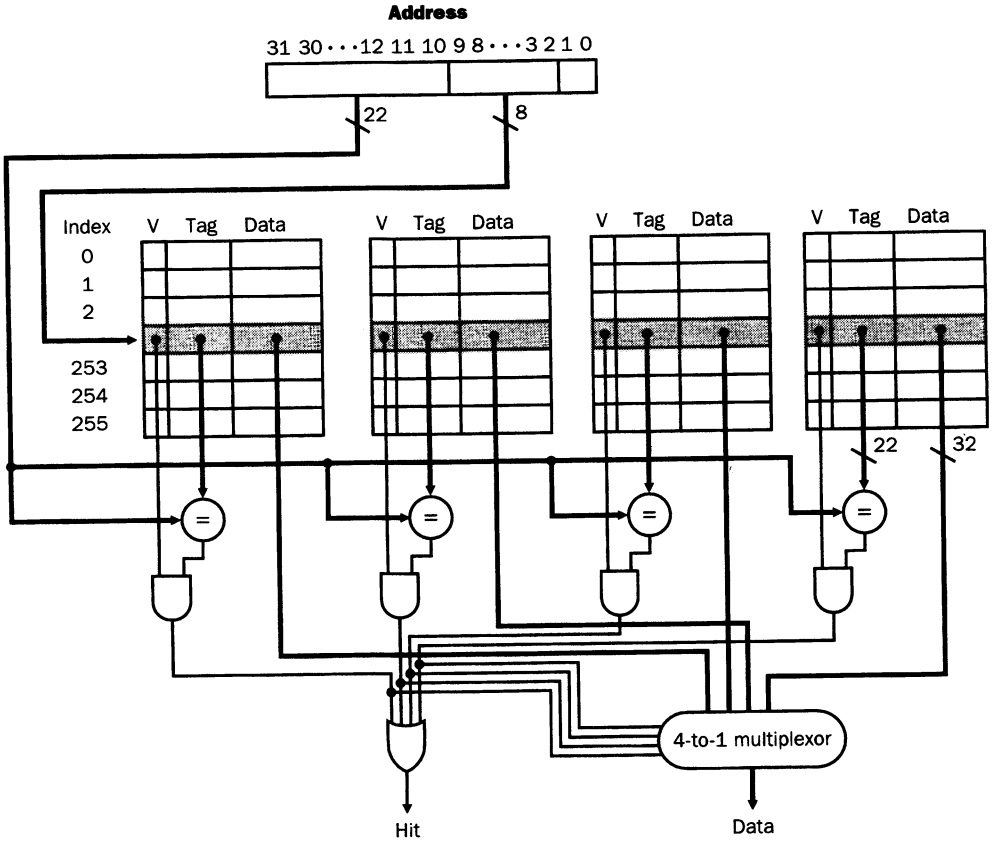

Start
1. Shift the Remainder register left 1 bit
2. Subtract the Divisor register from the left half of the Remainder register and place the result in the left half of the Remainder register
Test Remainder
Remainder ≥ 0
Remainder < 0
3a. Shift the Remainder register to the left, setting the new rightmost bit to 1
3b. Restore the original value by adding the Divisor register to the left half of the Remainder register and place the sum in the left half of the Remainder register. Also shift the Remainder register to the left, setting the new rightmost bit to 0
4th repetition?
No: < 4 repetitions
Yes: 4 repetitions
Done. Shift left half of Remainder right 1 bit

4 (a) A set-associative cache holds s sets of b blocks each. Any given block in memory goes to a well-defined set in the cache, depending on the block's address in memory:

$$\text{set} = (\text{block address}) \bmod s$$

Within the set, however, the block can go in any of the available places.

Now let's come to the tag. Assume I want to read a certain memory address. From the word address I can derive the block address and, from that, the correct set. But how do I know whether any of the block slots in that set actually contains the block I want? After all, a lot of blocks in memory map to that set. So, for each block in the cache, I must store the portion of the block's address that is not implied by the set number. This is the tag for that block. Without the tag, it is impossible to tell where a block in the cache actually came from.



(b) Byte in word = (byte address) mod (bytes per word) = 0x6717 mod 4 = 3 in all cases.

Word in block = (word address) mod (words per block) =
((byte address) div (bytes per word)) mod (words per block) =
(0x6717 div 4) mod (words per block) =
0x19c5 mod (words per block)

Block in set = arbitrary, could go anywhere

Block address in memory = (byte address) div (bytes per block)
= (byte address) div ((words per block) * (bytes per word))
= 0x6717 div ((words per block) * 4))

Set = (block address) mod (sets in cache)

Tag = (block address) div (sets in cache)

Cache size in bytes = (bytes per word) * (words per block) * (blocks per set)
* (sets in cache)

(i) A direct mapped cache is like a set-associative one with only one block per set, so we can pretend we have 4 "sets" in the above formulae. Block address = 0x6717 div 32 = 0x338

Byte	Word	Block	Set	Tag	Bytes-in-cache
3	5	0	n/a	0xce	128

(ii) Set-associative cache. Block address = 0x6717 div 16 = 0x671

Byte	Word	Block	Set	Tag	Bytes-in-cache
3	1	arbitrary	1	0x67	1024

(iii) A fully associative cache is like a set-associative one with just one large set. Block address = 0x6717 div 16 = 0x671

Byte	Word	Block	Set	Tag	Bytes-in-cache
3	1	arbitrary	n/a	0x671	256

Q4.b Numerical answer