DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Autumn Semester 2012-13 (2 hours)

EEE6031 Answers to Advanced Computer Architectures 6

1. a. *Describe a hypercube connection for a loosely-coupled multiprocessor.*

A hypercube connection between 2^n processors requires that each processor has n bi-directional links. Each processor is assigned a unique n bit address, and the links from a processor are connected to all other processors which possess an address whose Hamming distance is 1 from the address of the processor (there should be n in total). Data to be transferred between address a and b can be accomplished in a number of steps equal to the Hamming distance between the two addresses. **(4)**

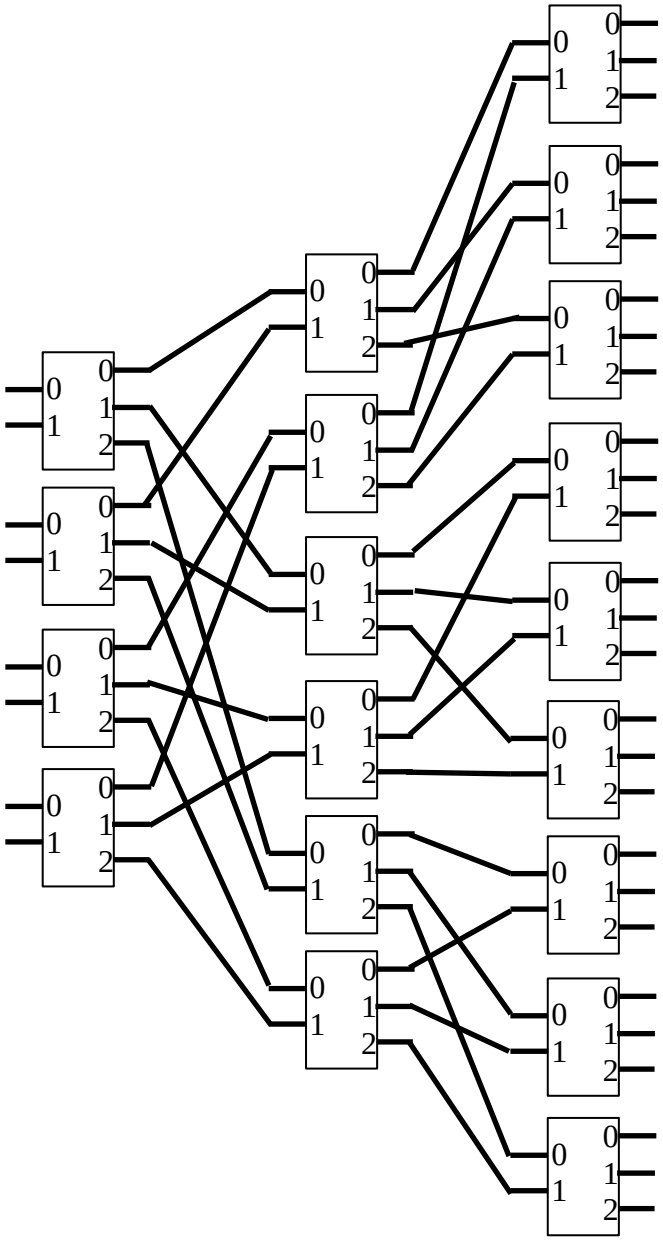
b. *What is the difference between packet- and circuit-switching for transmitting data in a network. What are the advantages and disadvantages of both schemes?*

One problem of a network where data has to cross multiple links to get from source to destination relates to the establishing of a link between source and destination - this is related to the average size of message. The fastest method is to establish the whole link from source to destination and to send data along it in one operation. This is fast, but tends to commit links and hold up other messages especially if the data blocks are large - this is circuit switching. Packet switching breaks messages up into small packets and transfers them individually across individual links, re-assembling the message at the destination. Links are only busy when it is reserved for a particular data packet. It is, however, generally more difficult to meet delay and delay variation specifications with packet switched data.

(4)

1. **c.** *Draw the block diagram for a 2:3:2 (s:f:l) Banyan network where the terms have their usual meaning.*

l=2 therefore two levels of interconnect.



(6)

d. *Derive an expression for the complexity of an s:f:l Banyan network.*

The final *s*:*f*:*l* Banyan network has f^l, $s \times f$ networks in the rightmost column and, therefore, it has f^{l+1} outputs and $s \times f^l$ inputs. These $s \times f^l$ inputs are the same in number as the outputs of the previous column of networks and, given that there are f outputs per Cross-bar network, there must be $s \times f^{l-1}$ Cross-bar networks in that column with $s^2 \times f^l$ inputs. By continuing this argument it can be shown that there are s^l, $s \times f$ networks in the leftmost column and that there are s^l+1 inputs to the network.

The total number of $s \times f$ networks in the *s*:*f*:*l* Banyan is:

$$f^l + s \times f^{l-1} + s^2 \times f^{l-2} \ldots + s^i \times f^{l-i} \ldots + s^{l-1} \times f + s^l = \sum_{i=0}^{l} s^{l-i} \times f^i$$

Thus, the complexity of the switch is:

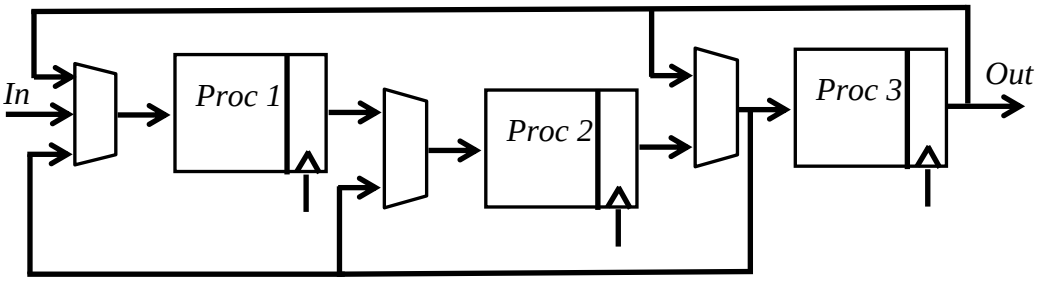
$$O\left(\sum_{i=0}^{l} s^{l-i+1} \times f^{i+1}\right)$$

and this can be compared to the complexity of a similar sized Cross-bar network

$$O\left(s^{l+1} \times f^{l+1}\right)$$

(6)

2. **a.** *A pipelined system appears as in **Figure 2**.*



The sequence of processing activities is:

In→Proc1→Proc2→Proc2→Proc3→Proc3→Proc1→Proc2→Proc3→Out

For this sequence:

i. *Draw the reservation table*

Time	Proc1	Proc2	Proc3
0	D1		
1		D1	
2	D2	D1	
3		D2	D1
4		D2	D1
5	D1		D2
6		D1	D2
7	D2		D1
8	D3	D2	
9		D3	D2
10	D4	D3	
11		D4	D3
12		D4	D3
13	D3		D4
14		D3	D4
15	D4		D3
16	D5	D4	
17		D5	D4

(4)

ii. *Calculate the collision vector*

Looking at Proc1, there is a gap of 4 timeslots between use and reuse,

Looking at Proc2, there are gaps of 0, 3, and 4 timeslots between use and reuse,

Looking at Proc3 there are gaps of 0, 2, and 3 timeslots between use and reuse.

Hence collision vector = 011101 **(2)**

iii. *Find the throughput of the pipeline*

The repeat pattern in the table shows that two datum can be entered every 8 timeslots. Hence, 0.25 datum/timeslot is the throughput. **(4)**

iv. *Find the utilisation of each processing block*

Proc1 is utilised 50% of the time, Proc2 is 75%, Proc3 is 75%. **(2)**

b. *You notice that the throughput can be improved by some minor changes to the pipeline. Show how, with some minor changes, the peformance can be improved and calculate a new throughput to show that performance has improved.*

There are various changes that could be made. For instance, putting two registers in series in the path feeding back from Proc3 to Proc1. In this case, the reservation table would change to:

Time	Proc1	Proc2	Proc3
0	D1		
1		D1	
2	D2	D1	
3		D2	D1
4	D3	D2	D1
5		D3	D2
6		D3	D2
7	D1		D3
8	D2	D1	D3
9	D3	D2	D1
10	D4	D3	D2
11		D4	D3
12	D5	D4	
13		D5	D4
14	D6	D5	D4
15		D6	D5
16		D6	D5
17			D6
18			D6

Yielding a throughput of 3 datum every 10 timeslots or 0.33datum/timeslot.
Hence and improvement of 33%.

(8)

3. a. **i.** *State Amdahl's Rule and identify its significance in understanding the benefits of parallelising any activity.*

Amdahl's Rule is

$$speedup = \frac{1}{(1-\alpha) + \frac{\alpha}{N}}$$

where α is the proportion of the task that can be parallelised (which must run in sequence with a proportion, 1-α that cannot be parallelised). For the part that can be parallelised the degree of parallelisation is *N*. **(4)**

ii. *Show how the basic Amdahl's rule can be extended to cope with a more generalised case of an activity where separate parts can be parallelised differently.*

If there are *M* parts to the activity where each part or proportion α_i can be parallelised with its own degree of parallelisation N_i then Amdahl's Rule can be extended to:

$$speedup = \frac{1}{\sum_{i=0}^{M-1} \frac{\alpha_i}{N_i}}$$ **(4)**

b. *A task running on a system consists of a bidirectional communication interface (running on a separate control processor) that is receiving a sequence of messages and farming each of the messages out to a number of separate data processors for processing. The results from each processor are transferred back to the communication interface and transmitted as a sequence of messages.*

Each received, input message takes 10μs to receive, decode, and farm out. Each transmitted, result message takes 1.5μs to collect, encode, and transmit. Each message can be processed in 170μs.

i. *Identify and justify your choice for the number of separate data processors (for processing the messages) that you would use in this system.*

The single control processor is receiving and sending messages via a duplex link and the limiting factor is the number of messages that are received. Consequently, a balanced system is created when the time taken to receive *N* messages is balanced by the time taken to process *N* messages on *M* data processors. Hence, *M*=17. **(4)**

ii. *What is the throughput of message for your chosen number of data processors?*

For 17 (or whatever number the student comes up with) data processors, the throughput of messages will, again, be dictated by the rate of receipt. That is 10^5 messages per second. **(2)**

iii. *What is the latency associated with each message?*

The latency is the time taken to receive, farm-out, process, collect and return a message and this is 10+170+1.5=182.5μs. **(2)**

iv. *A decision is made to alter the communication interface from bidirectional to unidirectional. How does this affect the design and performance of the system?*

The simplex communication link changes the system because all communications

use the same link. Consequently, the time taken to receive and transmit (assuming that these form the major part of the time) will be 11.5μs and now *M* would be 170/11.5=14.75 rounds up to 15 processors. Additionally, the throughput would be 8695 messages per second. **(4)**

4. a. *What characteristics make a memory technology amenable to use within a memory hierarchy where block data transfers are used to/from the memory?*

High speed – to complete first access into a random memory location,
High throughput – fast access to contiguous memory locations once an initial access has been undertaken (block transfers),
High density – to create large volumes of memory
Low cost,
Low power consumption **(4)**

b. *Describe, particularly, how SDRAM (the dominant memory used for main memory in a system is designed to be amenable for use within a memory hierarchy.*

SDRAM is DRAM with a synchronous wrapper around it allowing communication to be undertaken synchronously with the processor. The self-timed nature of accesses is hidden behind numbers of clock cycles of latency that the command interface must adhere to. Operations such a page-open, block read/write, and page close, refresh are separate operations allowing the processor to undertake split transactions. Once a page is opened it can be held open for multiple read/write operations. Block transfers are supported. SDRAMs normally consist of separate interleaved banks so that the delay in opening a page in one back can be hidden behind an operation on another bank. **(4)**

c. *Why is it important to use Read-Around-Write in the buffer systems that sit between the caches and main memory?*

The buffer that sits between the caches and main memory is important because it allows reads/writes to memory be queued and (in some cases) rearranged to make the most efficient use of the underlying memory. However, reads and writes are queued separately with reads having priority (this is read around write). The reason is that whilst the system will stall if the write queue overflows the system will stall when any read is delayed. This means that reads must be processed first. The only complication is that the write queue must be checked to ensure that the target of the read is not held in the write queue in which case this is the data that must be returned. **(4)**

d. *A memory sub-system has the following characteristics:*

Initial access time 5ns

Any subsequent access 5ns

Word width 32 bits

It is required that the memory sub-system, where it is used in the hierarchy, must be capable of supplying blocks of contiguous data at an average rate of at least 3 Gbytes/sec and that interleaving will be used to achieve this.

i. *Describe how interleaving works and how it helps.*

This memory technology does not seem to have a subsequent access time that is faster than the initial one and this means that we would need to use interleaving

to meet the throughput requirement. In this case, *N* banks of memory are interleaved (ie. Put in parallel) and accessed simultaneously, giving access to *N* words all together (at the expense of a bus that is *N* times wider) or a multiplexer to give the semblance of a faster interface. If the blocks are to be contiguous, this is done by ignoring the bottom *n* address bits (where *n* is $\log_2 N$) and addressing all of the memory banks using the same address **(4)**

ii. *Show that the memory be four-way interleaved to meet the data rate objective.*

The basic memory throughput rate is 4/5ns byte/s= 0.8Gbytes/s. If the memory is interleaved *N* ways then this improves the throughput to 0.8*N*Gbytes/sec. Consequently, if *N*=4 then throughput is 3.2GBytes/sec and this just meets the requirement. **(2)**

iii. *Does the interleaving place any limitations on the block/line size at this point in the hierarchy?*

Yes. Minimally, a block would be read as a single memory operation and this would equate to 4*N* bytes. Consequently, the block size is a multiple of 4*N*. **(2)**

iv. *The memory technology is changed so that the initial access time is 5ns but subsequent, contiguous access times are now 0.5ns. What is the minimum size of block transferred now to meet the data rate requirement – without using interleaving.*

To read a block of memory (4 bytes at a time) now takes 5ns for the first access and 0.5ns for each subsequent access. So the data rate for *N* reads would be $4N/(5+0.5N)$ Gbytes/sec. For this to be greater than or equal to 3 requires $2.5N \geq 15$. Therefore, $N \geq 15/2.5$ or $N \geq 6$. Therefore, the block size ≥ 24.

NLS