Indian Institute of Technology Jodhpur

Computer Organization (CS30002) End Semester Examination (Solution)

July-November, 2014 *Duration 3 Hours* *Date-24/11/2014*

Instructions:

- This question paper is in two pages.
- Read the questions carefully.
- Avoid seeking clarifications from the invigilators, unless it is very necessary.
- All questions are complete in every respect.
- Attempt all questions.

---

1. (a) Given RAM memory chips of 16 k X 8-bits each, how you will construct a 64 k X 8-bit memory system. The CPU has $A_0 \dots A_{15}$ address lines, and 8 data lines. Each memory chip has $A_0 \dots A_{13}$ address lines, 8-data lines, and a chip select line. Draw the the inter-connections for the memory system. [4]

Ans. The lines $A0 - A15$ are coming from the CPU. The $A14 - A15$ are input to an encoder, which gives four outputs $CE0 - CE3$. These chip enable lines are used for selection of the 1st 16k to last 16k of the 64k as shown in the figure 1. The table **??**shows the encode table.

A15	A14	CE0	CE1	CE2	CE3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

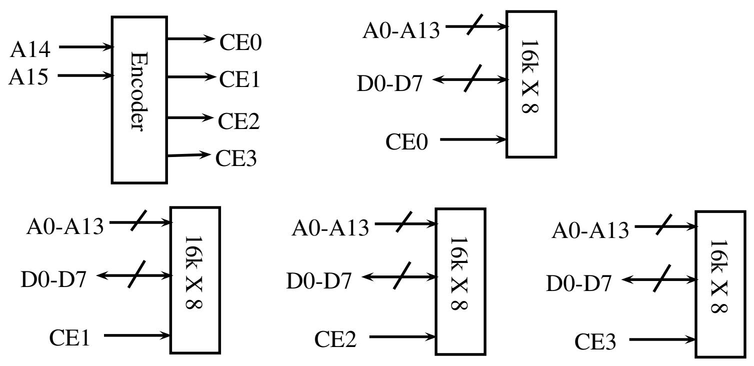

Figure 1: Constructing 64k from 16k RAMs.

(b) Consider a 32-bit RISC micro-processor with 32-bit data-bus and 32-bit address bus. The CPU operates at 1 GHz clock rate, and memory *load* and *store* instructions each take two CPU clock cycles. Memory-mapped IO is used and CPU supports both vectored and non-vectored interrupts, and DMA block-transfer with arbitrary block length. Typical interrupt response time is 10 CPU clock cycles. It is desired to add to the system, a disk drive with a data transfer rate of N-bits /sec. Estimate the maximum value that N can have for each of the following ways of controlling disk drive: [3+3]

i. Programmed IO using interrupts

ii. DMA

Show all you calculations and state the assumptions you have made for both hardware and software.

Ans. (i) The load store each takes 2 CPU cycles. One sec. has 1×10^9 cycles, so 10 interrupt cycle are negligible compared to that.

Since it is RISC processor, the execution is fast. So we assume that when one instruction is fetched other is executed. So net times for one data transfer in interrupt case is:

instruction fetch = 2 cycles

load data from IO to cpu = 2 cycles

store data from cpu to memory = 2 cycles

Total for one data transfer = 6 cycles. So no. of data transfer in 1 sec. = $10^9/6$ words. So the rate in bits per sec. is

$$\frac{32 \times 10^9}{6} = 5.3$$

Giga bits per sec. We have assumed that it is vectored interrupt so no time spent for transfer of control. Also, the time for delay of recognition of interrupt of 10 cycles is also neglected.

(ii) In DMA: One word is transferred per cycle. So transfer rate is 32 giga bits per sec.

2. (a) Consider a cache with a line size of 64 bytes. Assume that on average 30% of the lines in the cache are dirty (modified). A word consists of 8 bytes. Assume there is a 0.97 hit ratio. Compute the amount of main memory traffic, in terms of bytes per instruction for both *write-through* and *write-back* policies. Memory is read into cache one block at a time, with block size equal to line size. However, for write back, a single word can be written from cache to main memory. [4]

Ans. Consider the execution of 100 instructions. Under write-through, this creates 200 cache references (168 read references and 32 write references). On average, the read references result in (0.03) x 168 = 5.04 read misses. For each read miss, a line of memory must be read in, generating 5.04 x 8 = 40.32 physical words of traffic. For write misses, a single word is written back, generating 32 words of traffic. Total traffic: 72.32 words.

So for **write through:**

168*0.03*8 = 40.32 physical words of traffic

So total = 40.32 + 32 = 72.32 words of traffic.

for per instruction is = 72.32/100 = 0.7232

For **write back:**

For *write back*, 100 instructions create 200 cache references and thus 6 cache misses. Assuming 30% of lines are dirty, on average 1.8 of these misses require a line write before a line read. Thus, total traffic is (6 + 1.8) x 8 = 62.4 words.

So per instruction, it is = 62.4/100 = 0.624

(b) Design a cache memory system with 8-way 4 MB set associative cache memory, with line size and block size of 32-bytes. The size of main memory is 2 GB. Answer the following for this cache: [2+1+1+2]

i. How address mapping is done for a block in RAM with the line in cache?

ii. What are the sizes of: Tag field, offset, and set ?

iii. What is effect of increasing and decreasing the number of sets?

iv. What block replacement algorithms you would use for following? justify.

A. banking transactions (same algorithm for most transactions)

B. university computing (different algorithms)

Ans. For RAM: Let w is number of bits to represent address of a word in a block, so $w = 5$. 2GB $= 2^{31}$. s is block address bits, so $s = 31 - 5 = 26$.

For Cache: Cache size = 4 MB $= 2^{22}$. Number of lines $= 2^{22-5} = 2^{17}$. Number of sets $= 2^{17-5} = 2^{12}$, so $u = 12$.

A. mapping: set number $i = n \bmod 2^{14}$, n is block number and i is set number to which it is mapped.

B. tag=$s - u = 26 - 12 = 14$, offset = 3, set =12.

C. Deceasing moves sr boundary right (reducing u), increases associativity, increases tag field. Moving left increases number of sets, decreases associativity, decreases tag. When s equals to size of cache, it is direct mapped.

D. Block replacement: 1. for prog. LRU ok, for data FIFO ok. 2. Prog. LRU ok. For data any thing ok.

3. (a) Consider that in a CPU there are 8-interrupts, $I_0 \ldots I_7$. Design an interrupt priority system, such that I_j has higher priority than I_i, when $j > i$. There can be any number of interrupts active at a time but CPU will serve the highest priority. [4]

Ans. One of the approach is interrupt priority vector, as shown in figure 2.

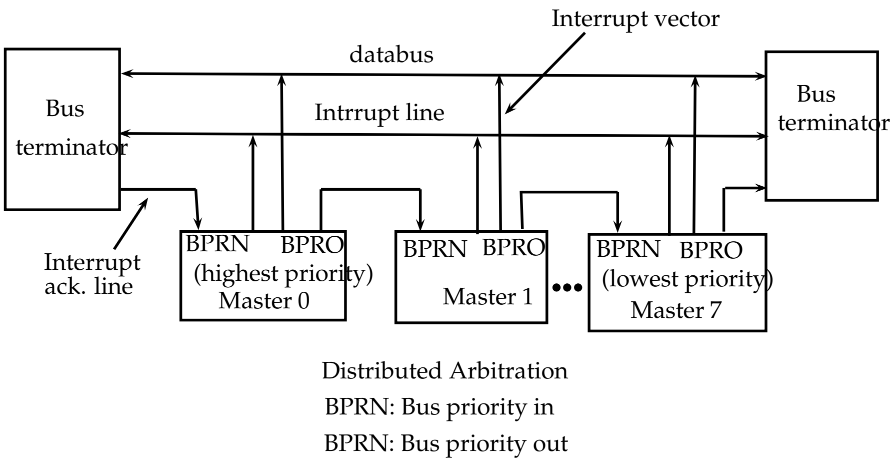

Figure 2: Interrupt priority scheme.

(b) An IO processor (IOP) controls the data transfer between main memory and set of IO devices with widely varying data transfer rates. The IOP can interleave (multiplex) transfers involving several IO devices provided that their combined effect does not exceed data transfer capacity of the system. The Data transfers are initiated by requests from the IO devices. A request is accepted by IOP only if it has sufficient spare capacity to service the requesting IO device. Write an algorithm to be run by the IOP to determine whether or not it can service a request from the next IO device. [6]

Ans. Say, there are n devices with data transfer capacity $d_0 \ldots d_{n-1}$. whether a device ready is indicated by a flag $f_i = 1$, and $f_i = 0$ indicates that device is not ready. Let the maximum capacity of IO transfer is D. To consider the device d_j for data transfer through IO processor, we allow it if

$$\sum_{i=0}^{n-1} f_i \leq D$$

Thus an algorithm can be :

```
1: while True do
2:   if $\sum_{i=0}^{n-1} f_i \leq D$ then
3:     set flag $f_j$ to true
4:     allow the device $f_j$ to transfer
5:   else
6:     set flag $f_j$ to false
7:   end if
8:   delay (few milli secs.)
9: end while
```

4. (a) Consider matrix multiplication operation on two matrices $A[4,5]$ and $B[5,6]$, producing a result matrix $C[4,6]$. The matrices are stored in RAM in row major order (each row is stored in consecutive memory locations). Design a pipeline to compute the resultant matrix. Answer the following: [2+2+2]

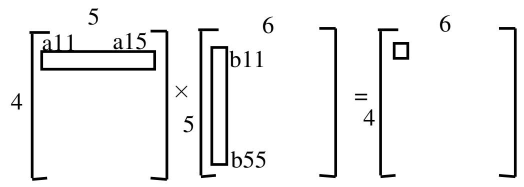

Figure 3: Matrices to multiply.

Ans. The figure 3 shows the matrices to be multiplied. The resultant cell c_{11} is given as;

$$c_{11} = a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} + a_{14} * b_{41} + a_{51} * b_{15}$$

i. How many pipeline segments you suggest ? Justify.

Ans. The figure 4 sows the pipeline.

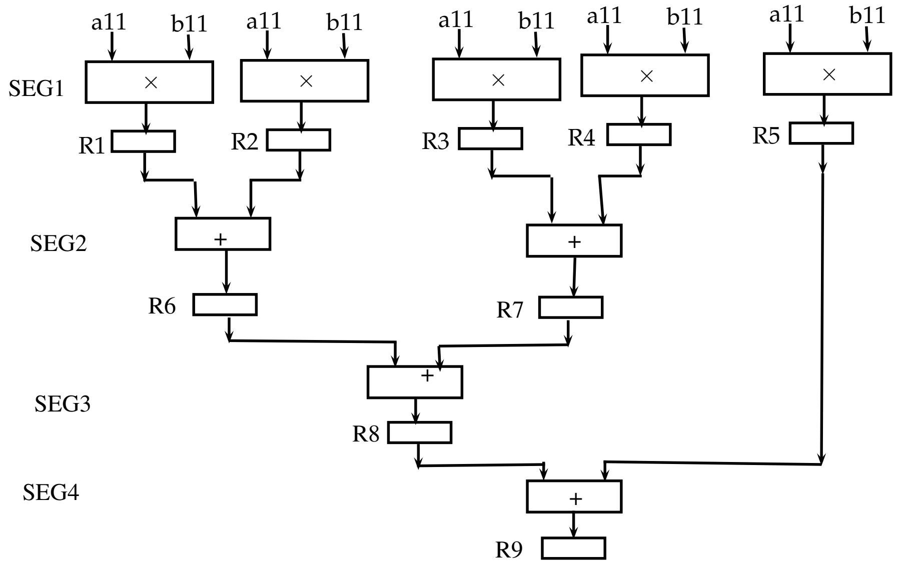

Figure 4: Pipeline to multiply matrices.

ii. What is instruction format and why?

Ans. The instructions are:

```
Load Reg, Adr
Add Reg, Reg, Reg
Mult Reg, Reg, Reg
Stor Adr, Reg
```

iii. How the resultant matrix is computed?

Ans. This is algorithm for matrix multiplication.

(b) A hypothetical time-sharing system has following properties. The system handles up to four concurrent jobs ($p_1 \dots p_4$), each requiring on the average of 5 secs of CPU time, which is allocated to the job in the time-slice manner with each slice of 50 msec. The CPU processes a job until an IO operation is required or time slice is over. The IO operation is *page swap* with secondary memory, and requires 100 msec. Find out the following performance measures: [2+2]

i. Mean response time (time to complete a job) when there are four active jobs.

Ans. The jobs are executed in time slices, i.e., each for 50 msec., i.e. p1 (50 msec),p2 (50 msec), p3 (50 msec), p4 (50 msec), p1 (50 msec), ..., until the finish in the same order. Thus each one will take 5 sec. only. The IO time needed is for loading each program in the begin (only), so 100 msec. for each. So total time = 20 msec. + 400 m.sec. So mean time = 5.1 msec.

ii. The mean fraction of time the CPU is idle.

Ans. CPU will be idle during the swap time, which is 0.40 sec. And, mean = 0.1 sec.

Following does not carry any marks and optional for attempting:
5. Rewrite these topics in preferential order of your liking (the best as first:)
Assembly language, RISC/CISC, interrupts/DMA, control design, Micro-programmed control, arithmetics, gate logic and flip-flops, cache, IO devices, Parallel computing.

BEST WISHES !!