Chapter 6. Memory System Design

Problem 6.1

The memory module uses 64 $4^M \times 1^b$ chips for 32MB of data and 8 $4^M \times 1^b$ chips for ECC. This allows 64 bits + 8 bits ECC to be accessed in parallel, forming a physical word.

T_{access}/module	=	120 ns
T_c	=	120 ns
T_{nibble}	=	40 ns up to four accesses
Physical word	=	64 bits + 8 bits ECC
Bus transit	=	20 ns (one way)
ECC	=	40 ns

a. Memory system access time

$$\begin{aligned} &= \text{Bus transit} + T_{\text{access}}/\text{module} + T_{\text{ECC}} + \text{Bus transit} \\ &= 20 + 120 + 40 + 20 \\ &= 200\,\text{ns} \end{aligned}$$

b. Maximum memory data bandwidth

Since we are allowed to have multiple buses and ECC units, we think of 4 memory modules as one very wide memory with 256 bits (64 bits × 4) wide datapath. That is, 256 bits can be fetched in parallel. The only limiting factor here is T_c for each module. For random access, each module cannot be accessed faster than 120 ns. So, the maximum bandwidth is $\frac{256 \text{ bits}}{120\,\text{ns}} = 267 \times 10^6 Bps = 254 MBps$.

c. From above, 256 bits can be fetched in parallel. We can use nibble mode for 4 consecutive accesses. Thus we can fetch $4 \times 256 = 1024$ bits every $120 + (4-1) \times 40 = 240$ ns. So the maximum bandwidth is $\frac{1024 \text{ bits}}{240\,\text{ns}} = 533 \times 10^6 Bps = 509 MBps$.

The low-order bits of the address are divided up as follows:

bits 0–2:	byte offset
bits 3–4:	module address
bits 5–6:	nibble address

Problem 6.2

a. For a page mode, the best organization will place a single page on a single module; this takes advantage of locality of reference as consecutive references to a single (2K) page will be able to take advantage of page mode. This is better than nibble mode, where only references to the same four words can take advantage of the optimized T_{nibble} time. Thus, the lower 11 bits of the word address are used as a (DRAM) page offset, and the address is broken up as follows:

bits 0–2: byte offset

bits 3–13: page offset

bits 14–15: module address

The advantage to interleaving is that accesses to different pages can be overlapped.

b. This system will perform significantly better than nibble mode for:
   1. Non-sequential access patterns.
   2. Access patterns that exhibit locality within a DRAM page.
   3. Access patterns that exhibit locality within multiple pages.
   4. Access patterns that sequentially traverse pages.

Problem 6.3

Hamming Code Design

Data size $(m) = 18$ bits

$2^k \geq m + k + 1$

Solve for k, and get $k = 5$

Total message length $= 18 + 5 = 23$ bits

Each correction bit covers its group bits (m)

k_1: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23

k_2: 2–3, 6–7, 10–11, 14–15, 18–19, 22–23

k_3: 4–7, 12–15, 20–23

k_4: 8–15

k_5: 16–23

1. Code for SEC (single error correction):

f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
k	k	m	k	m	m	m	k	m	m	m	m	m	m	m	k	m	m	m	m	m	m	m

The logic equation for each k is just XOR of each k's group bits.

2. Code for DEC (double error correction):

To detect double bit errors, we must add a final parity bit for the entire SEC code.

f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
k	k	m	k	m	m	m	k	m	m	m	m	m	m	m	k	m	m	m	m	m	m	m	k

Problem 6.4

- 40 MIPS
- 0.9 Instruction refs/Inst
- 0.3 Data read/Inst and 0.1 Data write/Inst
- $T_a = 200\,\text{ns}$
- $T_c = 100\,\text{ns}$

- Use open queue model

a. The allocation of modules to instruction and data

i) Memory modules for instruction

MAPS for instruction $= 0.9 \times 40 = 36$ MAPS

$\rho = \lambda_s/m \times T_c = 36 \times 10^6/m \times 100 \times 10^{-9} = 3.6/m$

Use $m = 8$, $\rho = .45$ So, 8 modules are necessary.

ii) Memory modules for data

MAPS for data $= .4 \times 40 = 16$ MAPS

$\rho = \frac{16 \times 10^6}{m} \times 10^{-7} = \frac{1.6}{m}$

Use $m = 4$, $\rho = .4$ So, 4 modules are necessary.

12 modules are necessary in total.

b. Effective T_w per reference (overall)

Assume this is $M_B/D/1$

$p = \frac{1}{m}$

$T_w = \frac{1}{\mu} \times \frac{\rho - p}{2(1-\rho)} = T_c \times \frac{\rho - \frac{1}{m}}{2(1-\rho)}$

For instruction memory,

$$T_w = 100 \times 10^{-9} \frac{.45 - .125}{2(1 - .45)} = 29.55\,\text{ns}$$

For data memory,

$$T_w = 100 \times 10^{-9} \frac{.4 - .25}{2(1 - .4)} = 12.5\,\text{ns}$$

Overall $T_w = \frac{.9 \times 29.55\,\text{ns} + .4 \times 12.5\,\text{ns}}{.4 + .9} = 24.3\,\text{ns}$

c. Queue size

For instruction memory, $Q_{ot} = \lambda \times T_w$ for Inst $= 36 \times 10^6 \times 29.55 \times 10^{-9} = 1.06$

For data memory, $Q_{ot} = \lambda \times T_w$ for data $= 16 \times 10^6 \times 12.5 \times 10^{-9} = .2$

Overall $Q_{ot} = 1.06 + .2 = 1.26$

d. Comparison to a single integrated I and D memory system

MAPS $= 1.3 \times 40$ MIPS $= 52$ MAPS

$\rho = \frac{\lambda_s}{m} \times T_c = \frac{52 \times 10^6}{m} \times 100 \times 10^{-9} = \frac{5.2}{m}$

Use $m = 16$, $\rho = .325$

$T_w = T_c \frac{\rho - \frac{1}{m}}{2(1-\rho)} = 100\,\text{ns}\ \frac{.325 - \frac{1}{16}}{2(1 - .325)} = 19.4\,\text{ns}$

$Q_{ot} = T_w \times \lambda = 19.4\,\text{ns} \times 52 \times 10^6$ per sec $= 1.009$

Type	Number of modules	T_w	Q_{ot}
Split	12	24.3 ns	1.26
Integrated	16	19.4 ns	1

Problem 6.5

Use $M_B/D/1$ closed queue model for realistic results

$T_c = 100\,\text{ns}, T_a = 120\,\text{ns}$

Two references to memory in each memory cycle: $n = 2$

Eight interleaved memory modules: $m = 8$

a. Expected waiting time

$\rho_a = 1 + \frac{2}{8} - \frac{1}{2\times 8} - \sqrt{(1 + \frac{2}{8} - \frac{1}{2\times 8})^2 - \frac{2\times 2}{8}} = .233$

$T_w = T_c \frac{\rho_a - \frac{1}{m}}{2(1-\rho_a)} = 100\,\text{ns}\frac{.233-\frac{1}{8}}{2(1-.233)} = 7.04\,\text{ns}$

b. Total access time

$T_a + T_w = 120\,\text{ns} + 7.04\,\text{ns} = 127.04\,\text{ns}$

c. Mean total number of queued (waiting) requests

$n - B = n - m \times \rho_a = 2 - 8 \times .233 = .136$

d. Offered memory bandwidth

Offered $= n/T_c = 2/100\,\text{ns} = 20$ MAPS

e. Achieved memory bandwidth

$B(m,n)/T_c = 8 \times .233/100\,\text{ns} = 18.64$ MAPS

f. Achieved bandwidth using Strecker's model

$B(m,n) = m(1 - (1 - \frac{1}{m})^n) = 8(1 - (1 - \frac{1}{8})^2) = 1.875$

Bandwidth $= \frac{B}{T_c} = \frac{1.875}{100\times 10^{-9}} = 18.75$ MAPS

Problem 6.7

Integrated Memory with $m = 8$

$C_P = 2$ (Instruction source and data source; assumes no independent writes from a data buffer. If data buffer is assumed, then $C_p = 3$.)

$Z = C_P \times \frac{T_c}{\Delta T} = 3 \times \frac{100(\text{ns})}{1/(40\times 10^6)} = 12$

$\delta = \frac{n}{Z} = \frac{1.3}{12} \times \frac{100(\text{ns})}{1/(40\times 10^6)} = .433$

$$\begin{aligned} B(m,n,\delta) &= m + n - \tfrac{\delta}{2} - \sqrt{(m + n - \tfrac{\delta}{2})^2 - 2nm} \\ &= 8 + 5.2 - \tfrac{.433}{2} - \sqrt{(8 + 5.2 - \tfrac{.433}{2})^2 - 2 \times 5.2 \times 8} \\ &= 3.74 \end{aligned}$$

$\text{Perf}_{\text{ach}} = \frac{3.74}{5.2} \times 40$ MIPS $= 28.8$ MIPS.

Problem 6.8

$T_a = 200\,\text{ns}$, $T_c = 100\,\text{ns}$, $m = 8$, $T_{\text{bus}} = 25\,\text{ns}$, $L = 16$; Copyback with write allocate.

a. Compute $T_{\text{line.access}}$.

$L > m$ and $T_c < m \times T_{\text{bus}}$

$$\begin{aligned} T_{\text{line.access}} &= T_a + (L-1) \times T_{\text{bus}} \\ &= 200 + 15 \times 25 \\ &= 575\,\text{ns} \end{aligned}$$

b. Repeat for $m = 2$ and $m = 4$.

$m = 2, m < L$ and $T_c > m \times T_{\text{bus}}$

$$\begin{aligned} T_{\text{line.access}} &= T_a + T_c(\lceil \tfrac{L}{m} \rceil) + T_{\text{bus}} \times ((L-1) mod\ m) \\ &= 200 + 100(7) + 25(1) \\ &= 925\,\text{ns} \end{aligned}$$

$m = 4, m < L$ and $T_c \leq m \times T_{\text{bus}}$

$$\begin{aligned} T_{\text{line.access}} &= T_a + (L-1)T_{\text{bus}} \\ &= 200 + 15(25) \\ &= 575\,\text{ns} \end{aligned}$$

c. Nibble mode is now introduced:

$T_{\text{nibble}} = 50\,\text{ns}, m = 2, v = 4, T_v = 50\,\text{ns}$

$$\begin{aligned} T_{\text{line.access}} &= T_a + T_c(\lceil \tfrac{L}{m \times v} \rceil - 1) + T_{\text{bus}}(L - \tfrac{L}{m \times v}) \\ &= 200 + 100(1) + 25(16-2) \\ &= 650\,\text{ns} \end{aligned}$$

Problem 6.9

CBWA with $w = .5$, $T_a = 200\,\text{ns}$, $T_c = 100\,\text{ns}$, $m = 8$, $T_{\text{bus}} = 25\,\text{ns}$, $L = 16$

a. Unbuffered line transfer starting at line address

$T_{\text{m.miss}} = (1+w) \times T_{\text{line.access}} = 1.5 \times 575\,\text{ns} = 863\,\text{ns}$

$T_{\text{c.miss}} = (1+w) \times T_{\text{line.access}} = 1.5 \times 575\,\text{ns} = 863\,\text{ns}$

$T_{\text{busy}} = 0\,\text{ns}$

b. Write buffer, line transfer starting at line address

$T_{\text{m.miss}} = (1+w) \times T_{\text{line.access}} = 1.5 \times 575\,\text{ns} = 863\,\text{ns}$

$T_{\text{c.miss}} = T_{\text{line.access}} = 575\,\text{ns}$

$T_{\text{busy}} = w \times T_{\text{line.access}} = 288\,\text{ns}$

c. Access first word

$T_{\text{m.miss}} = (1+w) \times T_{\text{line.access}} = 1.5 \times 575\,\text{ns} = 863\,\text{ns}$

$T_{\text{c.miss}} = T_a = 200\,\text{ns}$

$T_{\text{busy}} = T_{\text{m.miss}} - T_{\text{c.miss}} = 663\,\text{ns}$

Problem 6.13

A processor without a cache accesses every t-th element of a k element vector. Each element is 1 physical word. Assuming $T_a = 200\,\text{ns}$, $T_c = 100\,\text{ns}$, and $T_{\text{bus}} = 25\,\text{ns}$, plot the average access time per element for an 8-way, low-order interleaved memory for $t = 1$ to 12 and $k = 100$.

Figure 2: Problem 6-13.

Problem 6.14

$\lambda_s = 75$ MAPS and $T_s = 100\,\text{ns}$

$\rho = \frac{\lambda_s}{m} T_s = \frac{75 \times 10^6}{m} 100 \times 10^{-9} = \frac{7.5}{m}$

Since ρ should be around .5, use $m = 16$. Then, $\rho = \frac{7.5}{16} = .469$.

$Q_o = \frac{\rho(\rho - p)}{2(1-\rho)} = \frac{.469(.469 - \frac{1}{16})}{2(1-.469)} = .1795$

$Q_{ot} = 16 \times .1795 = 2.87$

a. Using Chebyshev

   Prob (q > BF) $\leq .01$

   Prob (q $\geq$ BF +1) $\leq .01$

   $\frac{Q_o}{BF+1} \leq .01$

   BF $+1 \geq \frac{Q_o}{P} = \frac{.1795}{.01} = 17.95 \approx 18$

   BF ≥ 17

   Total BF (TBF) $= 17 \times 16 = 272$

b. Using M/M/1

Prob (overflow) $\leq .01$

Solve for $\rho^{(\mathrm{TBF}/m)+2} = .01$

$.469^{(\mathrm{TBF}/m)+2} = .01$

TBF ≈ 66

Problem 6.15

You are to design the memory for a 50 MIPS processor $(1\tilde{/}\mathrm{I})$ with 1 instruction and 0.5 data references per instruction. The memory system is to be 16MB. The physical word size is 4B. You are to use 1M x 1b chips with $\mathrm{T}_c = 40\,\mathrm{ns}$. Draw a block diagram of your memory including address, data, and control connections between the processor, DRAM controller, and the memory. Detail what each address bit does. If $\mathrm{T}_a = 100\,\mathrm{ns}$, what are the expected memory occupancy, waiting time, total access time, and total queue size? Discuss the applicability of the Flores model in the analysis of this design?

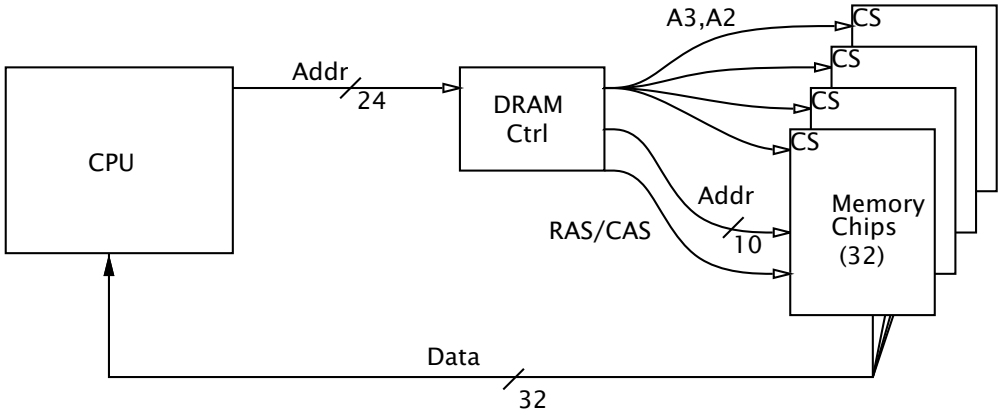

Figure 3: Problem 6-15.

$$
\begin{aligned}
\lambda &= 75MAPS \\
\mu &= m/Tc \\
\rho &= \lambda/\mu = 3/m \\
m &= 4 \\
\rho &= 0.75 \\
T_w &= T_c \frac{\rho - 1/m}{2(1-\rho)} = T_c = 40\,\mathrm{ns} \\
Q_o &= \frac{\rho(\rho - 1/m)}{2(1-\rho)} = 0.75 \\
Q_{o-t} &= 3
\end{aligned}
$$

Because of the high occupancy of the memory, the Flores model is not particularly well suited to this design.

Problem 6.17

IF/cycle = .5

DF/cycle = .3

DS/cycle = .3

$m = 8$ and $T_s = 100\,\text{ns}$(memory)

Processor cycle time $(\Delta T) = 20\,\text{ns} \rightarrow 50$ MIPS

$\lambda = .9 \times 50 \times 10^6 = 45$ MAPS

$n = (.5 + .3 + .1) \times \frac{100}{20} = 4.5$

$z = 3 \times \frac{100}{20} = 15$

$\delta = \frac{n}{z} = \frac{4.5}{15} = .3$

$\rho = \frac{n}{m} = \frac{4.5}{8} = .563$

$$\begin{aligned}
B(m,n,\delta) &= 8 + 4.5 - \frac{.3}{2} - \sqrt{(8 + 4.5 - \frac{.3}{2})^2 - 2 \times 8 \times 4.5} \\
&= 3.377 \\
Bw &= \frac{3.377}{10^{-7}} = 33.77 MAPS \\
\rho_a &= \frac{B}{m} = \frac{3.377}{8} = .422 \\
\text{MIPS}_{\text{ach}} &= \frac{\rho_a}{\rho} 50 \text{ MIPS} = \frac{.422}{.563} = 37.48 \text{ MIPS} \\
T_w &= \frac{n - B}{B} T_s = \frac{4.5 - 3.377}{3.377} \times 100(\text{ ns}) = 33.25(\text{ ns}) \\
Q_{ct} &= n - B = 4.5 - 3.377 = 1.123
\end{aligned}$$

Problem 6.18

a. Line size = 16B

This is already calculated in study 6.3. $\text{Perf}_{\text{rel}} = .78$

b. Line size = 8B

Miss rate = .07

L = $8B/4B = 2$

$$\begin{aligned}
T_{\text{line access}} &= T_a + T_c(\left\lceil \frac{L}{m} \right\rceil - 1) + T_{\text{bus}}((L-1) \bmod m) \\
&= 120 + 100(1-1) + 40((2-1) \bmod 2) \\
&= 120 + 40 = 160\,\text{ns} \\
T_{\text{m.miss}} &= (1+w) T_{\text{line access}} = 1.5 \times 160\,\text{ns} = 240\,\text{ns} \\
\text{Perf}_{\text{rel}} &= \frac{1}{1 + f\lambda_p T_{\text{m.miss}}} \\
&= \frac{1}{1 + .07 \times \frac{1}{40 \times 10^{-9}} \times 240 \times 10^{-9}} \\
&= .704
\end{aligned}$$

c. Line size = 32B

Miss rate = .02

L = $\frac{32B}{4B} = 8$

$$\begin{aligned} T_{\text{line access}} &= 120 + 100(\left\lceil \frac{8}{2} \right\rceil - 1) + 40((8-1) \bmod 2) \\ &= 460 \\ T_{\text{m.miss}} &= (1+w) \times 460\,\text{ns} = 1.5 \times 460\,\text{ns} = 690\,\text{ns} \\ \text{Perf}_{\text{rel}} &= \frac{1}{1 + .02 \times 1/(40 \times 10^{-9}) \times 690 \times 10^{-9}} = .743 \end{aligned}$$

In conclusion, the cache with 16B line size shows the best performance.