真理大學107學年度第1學期 期末複習考試 參考解答

考 試 科 目	計算機結構	■大學日間 □大學進學 □碩士班 □碩士在職班	資訊工程學系 3 年 B 班	命題老師	林熙中
考 試 日 期	12 月 28 日星期五 第 2~4 節	附答案紙□是 ■否 列印大小■A4□B4	試卷別■單一□A 卷□B 卷	印刷 份數	60
姓 名		學 號		序 號	

1. To enhance data reliability, a memory system uses Hamming error correction code to detect and correct one bit-error. (*16 points*)

   (a) Suppose 1010101 is the word to be written into memory. What is the resulting bit-sting stored in memory.

   (b) Now consider the read operation of the memory system. What is the correct data word embedded in the bit-string 00110001100 retrieved from memory? Will any error be detected during this read operation?

Sol:

(a) i. Firstly, 7 data bits require 4 check bits since $2^3 \ngeq 7+3+1$ but $2^4 \geq 7+4+1$.

ii. Secondly, arrange the 7 data bits D7∼D1 and the 4 check bits C8, C4, C2, and C1 as follows.

Position$^{(10)}$	11	10	9	8	7	6	5	4	3	2	1
Position$^{(2)}$	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D7	D6	D5		D4	D3	D2		D1		
Check bit				C8				C4		C2	C1

Those positions whose numbers are powers of 2 are designated as check bits.
C8 = D7 ⊕ D6 ⊕ D5
C4 = D4 ⊕ D3 ⊕ D2
C2 = D7 ⊕ D6 ⊕ D4 ⊕ D3 ⊕ D1
C1 = D7 ⊕ D5 ⊕ D4 ⊕ D2 ⊕ D1

iii. Given data word 1010101, we can calculate check bits as follows.

Position$^{(10)}$	11	10	9	8	7	6	5	4	3	2	1
Position$^{(2)}$	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D7 1	D6 0	D5 1		D4 0	D3 1	D2 0		D1 1		
Check bit				C8 0				C4 1		C2 1	C1 1
Data stored	1	0	1	0	0	1	0	1	1	1	1

C8 = D7 ⊕ D6 ⊕ D5 = 1 ⊕ 0 ⊕ 1 = 0
C4 = D4 ⊕ D3 ⊕ D2 = 0 ⊕ 1 ⊕ 0 = 1
C2 = D7 ⊕ D6 ⊕ D4 ⊕ D3 ⊕ D1 = 1 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 = 1
C1 = D7 ⊕ D5 ⊕ D4 ⊕ D2 ⊕ D1 = 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 = 1

iv. Finally, the data word stored is $101\underline{0}010\underline{1}1\underline{11}$

(b) i. Firstly, recalculate check bits as follows.

Position$^{(10)}$	11	10	9	8	7	6	5	4	3	2	1
Position$^{(2)}$	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data stored	D7 0	D6 0	D5 1	C8 1	D4 0	D3 0	D2 0	C4 1	D1 1	C2 0	C1 0

C8 = D7 ⊕ D6 ⊕ D5 = 0 ⊕ 0 ⊕ 1 = 1
C4 = D4 ⊕ D3 ⊕ D2 = 0 ⊕ 0 ⊕ 0 = 0
C2 = D7 ⊕ D6 ⊕ D4 ⊕ D3 ⊕ D1 = 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1
C1 = D7 ⊕ D5 ⊕ D4 ⊕ D2 ⊕ D1 = 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 = 0

ii. Compare (by exclusive-or) recalculated check bits with those directly read from memory.

$$\begin{array}{rcccc l}
 & \text{C8} & \text{C4} & \text{C2} & \text{C1} & \\
\text{recalculated} & 1 & 0 & 1 & 0 & \\
\text{read from memory} \oplus & 1 & 1 & 0 & 0 & \\
\hline
\text{syndrome word} & 0 & 1 & 1 & 0 & \Rightarrow \text{position 6} \Rightarrow \text{error in data bit D3}
\end{array}$$

iii. Output the 7-bit data word 0010101.

2. By adding one extra parity bit, Hamming error correction code is capable of detecting a double-bit error and correcting a single-bit error. (For simplicity, assume that the extra parity bit is placed as the left-most bit.) (*8 points*)

   (a) Suppose 1010101 is the word to be written into memory. What is the resulting bit-sting stored in memory.

   (b) Now consider the read operation of the memory system. What is the correct data word embedded in the bit-string 000110001100 retrieved from memory? Will any error be detected during this read operation?

Sol:

(a) i. Firstly, 7 data bits require 4 check bits since $2^3 \ngeq 7+3+1$ but $2^4 \geq 7+4+1$.

ii. Secondly, arrange the 7 data bits D7∼D1, the 4 check bits C8, C4, C2, C1, and an extra parity bit C_{all} (for double-bit error detection) as follows.

Position$^{(10)}$	12	11	10	9	8	7	6	5	4	3	2	1
Position$^{(2)}$	—	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit		D7	D6	D5		D4	D3	D2		D1		
Check bit	C_{all}				C8				C4		C2	C1

Those positions whose numbers are powers of 2 are designated as check bits.
$\mathrm{C8} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D5}$
$\mathrm{C4} = \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D2}$
$\mathrm{C2} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D1}$
$\mathrm{C1} = \mathrm{D7} \oplus \mathrm{D5} \oplus \mathrm{D4} \oplus \mathrm{D2} \oplus \mathrm{D1}$
$\mathrm{C}_{all} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D5} \oplus \mathrm{C8} \oplus \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D2} \oplus \mathrm{C4} \oplus \mathrm{D1} \oplus \mathrm{C2} \oplus \mathrm{C1}$

iii. Given data word 1010101, we can calculate check bits as follows.

Position$^{(10)}$	12	11	10	9	8	7	6	5	4	3	2	1
Position$^{(2)}$	—	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit		D7 1	D6 0	D5 1		D4 0	D3 1	D2 0		D1 1		
Check bit	C_{all} 1				C8 0				C4 1		C2 1	C1 1
Data stored	1	1	0	1	0	0	1	0	1	1	1	1

$\mathrm{C8} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D5} = 1 \oplus 0 \oplus 1 = 0$
$\mathrm{C4} = \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D2} = 0 \oplus 1 \oplus 0 = 1$
$\mathrm{C2} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D1} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$
$\mathrm{C1} = \mathrm{D7} \oplus \mathrm{D5} \oplus \mathrm{D4} \oplus \mathrm{D2} \oplus \mathrm{D1} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1$
$\mathrm{C}_{all} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D5} \oplus \mathrm{C8} \oplus \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D2} \oplus \mathrm{C4} \oplus \mathrm{D1} \oplus \mathrm{C2} \oplus \mathrm{C1} = 1$

iv. Finally, the data word stored is $\underline{1}101\underline{0}010\underline{1}1\underline{1}\underline{1}$

(b) i. Firstly, recalculate check bits as follows.

Position$^{(10)}$	12	11	10	9	8	7	6	5	4	3	2	1
Position$^{(2)}$	—	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data stored	C_{all} 0	D7 0	D6 0	D5 1	C8 1	D4 0	D3 0	D2 0	C4 1	D1 1	C2 0	C1 0

$\mathrm{C8} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D5} = 0 \oplus 0 \oplus 1 = 1$
$\mathrm{C4} = \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D2} = 0 \oplus 0 \oplus 0 = 0$
$\mathrm{C2} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D1} = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$
$\mathrm{C1} = \mathrm{D7} \oplus \mathrm{D5} \oplus \mathrm{D4} \oplus \mathrm{D2} \oplus \mathrm{D1} = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0$

ii. Compare (by exclusive-or) recalculated check bits with those directly read from memory.

$$\begin{array}{rccccl} & \mathrm{C8} & \mathrm{C4} & \mathrm{C2} & \mathrm{C1} & \\ \text{recalculated} & 1 & 0 & 1 & 0 & \\ \text{read from memory } \oplus & 1 & 1 & 0 & 0 & \\ \hline \text{syndrome word} & 0 & 1 & 1 & 0 & \Rightarrow \text{position 6} \Rightarrow \text{error in data bit D3} \end{array}$$

iii. Fix the error in check bit D3.

iv. Compute $\mathrm{C}'_{all} = \mathrm{D7} \oplus \mathrm{D6} \oplus \mathrm{D5} \oplus \mathrm{C8} \oplus \mathrm{D4} \oplus \mathrm{D3} \oplus \mathrm{D2} \oplus \mathrm{C4} \oplus \mathrm{D1} \oplus \mathrm{C2} \oplus \mathrm{C1} = 1$.
Since $\mathrm{C}'_{all} \neq \mathrm{C}_{all}$, we conclude that there is a double-bit error and it can not be fixed.

v. Finally, output an error signal.

3. Answer the following questions concisely and precisely? (*8 points*)

   (a) Explain the technique of "CAV" used in magnetic disks. What are the main advantage and disadvantage.

   (b) Explain the technique of "CLV" used in optical disks. What are the main advantage and disadvantage.

Sol:

(a) A bit near the center of a rotating disk travels past a fixed point (such as a read/write head) slower than a bit on the outside. Therefore, some way must be found to compensate for the variation in speed so that the head can read all the bits at the same rate. This can be done by increasing the spacing between bits of information recorded in segments of the disk. The information can then be scanned at the same rate by rotating the disk at a fixed speed, known as the constant angular velocity (CAV). The disk is divided into a number of pie-shaped sectors and into a series of concentric tracks. The advantage of using CAV is that individual blocks of data can be directly addressed by track and sector. To move the head from its current location to a specific address, it only takes a short movement of the head to a specific track and a short wait for the proper sector to spin under the head. The disadvantage of CAV is that the amount of data that can be stored on the long outer tracks is the only same as what can be stored on the short inner tracks.

(b) To achieve greater capacity, CDs and CD-ROMs do not organize information on concentric tracks. Instead, the disk contains a single spiral track, beginning near the center and spiraling out to the outer edge of the disk. Sectors near the outside of the disk are the same length as those near the inside. Thus, information is packed evenly across the disk in segments of the same size and these are scanned at the same rate by rotating the disk at a variable speed. The pits are then read by the laser at a constant linear velocity (CLV). The disk rotates more slowly for accesses near the outer edge than for those near the center. Thus, the capacity of a track and the rotational delay both increase for positions nearer the outer edge of the disk.

4. Consider a magnetic disk drive with 4 surfaces, 256 tracks per surface, and 128 sectors per track. Sector size is 1KB and the drive rotates at 3000rpm. Its average seek time is 4ms, while track-to-track access time is 2ms. Successive tracks in a cylinder can be read in turn without heads movement and additional rotational delay, but can not be read simultaneously. (*20 points*)

   (a) What is the disk capacity?

   (b) What is the average access time?

   (c) Estimate the time required to transfer a 800KB file which is optimally organized.

   (d) Estimate the time required to transfer a 800KB file where sectors are distributed randomly over the disk.

   (e) What is burst transfer rate (the highest speed at which data can be transferred)?

Sol:

(a) $4 \times 256 \times 128 \times 1\text{KB} = 128\text{MB}$

(b) Average access time = average seek time + rotational delay

- Average seek time = 4ms
- 3000rpm = 50rps $\Rightarrow$ 1 revolution requires $\frac{1}{50}$s = 20ms $\Rightarrow$ rotational delay = $\frac{20}{2}$ = 10ms
- Average access time = 4+10 =14ms

(c) A track consists of 128 sectors, sector size is 1KB $\Rightarrow$ track capacity is 128KB
$\Rightarrow$ a 800KB file requires $\frac{800\text{KB}}{128\text{KB}} = 6.25$ tracks.
A cylinder consists of 4 tracks $\Rightarrow$ a 800KB file requires 1 cylinder plus 2.25 tracks.

- The 1st cylinder requires 4ms seek time, 10ms rotational delay, followed by 4 revolutions.
  $\Rightarrow 4 + 10 + (4 \times 20)$
- The 2nd cylinder requires 2ms seek time, 10ms rotational delay, followed by 2.25 revolutions.
  $\Rightarrow 2 + 10 + (2.25 \times 20)$
- Totally, $[4 + 10 + (4 \times 20)] + [2 + 10 + (2.25 \times 20)] = 151\text{ms}$

(d) A 800KB file requires $\frac{800\text{KB}}{1\text{KB}}$ = 800 sectors.

- To read a random sector, it requires 4ms seek time, 10ms rotational delay, followed by $\frac{1}{128}$ revolutions
  $\Rightarrow 4 + 10 + (\frac{1}{128} \times 20)$
- 800 sectors require $800 \times [4 + 10 + (\frac{1}{128} \times 20)] = 11325\text{ms}$

(e) Burst transfer rate = (revolutions/sec) $\times$ (sectors/revolution) $\times$ (bytes/sector)
$= \frac{3000}{60} \times 128 \times 1\text{KB} \approx 6.4\text{MB/s}$

5. Find the sum of 01111001 and 00010111 assuming numbers are represented in (*8 points*)

   (a) sign-magnitude representation;

   (b) twos complement;

(a) overflow

	sign	magnitude
	$+ \Leftarrow 0$	1111001
$+$	$+ \Leftarrow 0$	0010111
	1	0010000
	$\uparrow$	
	overflow	

(b) overflow

	01111001
$+$	00010111
	10010000
	$\uparrow$
	overflow

6. How to expend the bit length of an integer in twos complement representation? Explain why it works. (*8 points*)

Sol:
Move the sign bit to the new leftmost position and fill in with copies of the sign bit. For positive numbers, fill in with zeros, and for negative numbers, fill in with ones.
For positive numbers, this rule clearly works. For negative numbers, this rule also works because this is the only way to keep the value unchanged. (Note: You can find more detail in Ch10.2 “Range Extension”.)

7. Given $x = 10011$ and $y = 11101$ in twos complement notation, do the following arithmetic.

   (a) $x - y$. (*4 points*)

   (b) $x \times y$. (Use the Booth algorithm and you should describe how registers are changed.) (*8 points*)

   (c) $x \;/\; y$. (Use the algorithm described in Sect.10.3 and describe how registers are changed.) (*8 points*)

(a) $x - y =$ overflow

	10011
$-$	11101
	$\Downarrow$
	10011
$+$	00011
	10110

(b) $x \times y = 0000100111$

A	Q	Q_{-1}	M	Count		
00000	11101	0	10011	5	Initial values	
01101	11101	0	10011	5	A=A-M	$\Leftarrow Q_0Q_{-1} = 10$
00110	11110	1	10011	4	Arithmetic shift right	
11001	11110	1	10011	4	A=A+M	$\Leftarrow Q_0Q_{-1} = 01$
11100	11111	0	10011	3	Arithmetic shift right	
01001	11111	0	10011	3	A=A-M	$\Leftarrow Q_0Q_{-1} = 10$
00100	11111	1	10011	2	Arithmetic shift right	
00010	01111	1	10011	1	Arithmetic shift right	$\Leftarrow Q_0Q_{-1} = 11$
00001	00111	1	10011	0	Arithmetic shift right	$\Leftarrow Q_0Q_{-1} = 11$
00001	00111	$\Leftarrow$ The product is in the A and Q registers.				

(c) $x \;/\; y = 00100 \ldots 11111$

A	Q	M	Count	
11111	10011	11101	5	Initial values
11111	$0011\underline{0}$	11101	5	Shift left
00010	$0011\underline{0}$	11101	5	A=A-M
11111	$0011\underline{0}$	11101	4	A=A+M (Restore)
11110	$011\underline{00}$	11101	4	Shift left
00001	$011\underline{00}$	11101	4	A=A-M
11110	$011\underline{00}$	11101	3	A=A+M (Restore)
11100	$11\underline{000}$	11101	3	Shift left
11111	$11\underline{000}$	11101	3	A=A+M
11111	$11\underline{001}$	11101	2	Set $Q_0 = 1$
11111	$1\underline{0010}$	11101	2	Shift left
00010	$1\underline{0010}$	11101	2	A=A-M
11111	$1\underline{0010}$	11101	1	A=A+M (Restore)
11111	$\underline{00100}$	11101	1	Shift left
00010	$\underline{00100}$	11101	1	A=A-M
11111	$\underline{00100}$	11101	0	A=A+M (Restore)
11111	00100			

The remainder is in the A register; i.e., 11111. The quotient is in the Q register; i.e., 00100.

8. Consider a reduced 16-bit floating point format, where the left most bit is a sign bit, the following 9 bits represent the exponent and the other 6 bits at the right-hand side are for the significand. For simplicity, we assume that values are normalized and the exponent is coded in biased representation. Please answer the following questions. (*16 points*)

   (a) Express $(-121/16)_{10}$ in this format.

   (b) What is the equivalent decimal value for $(1100\ 0011\ 0011\ 1100)_2$?

   (c) What is the smallest positive number (in decimal) that can be coded with this format?

   (d) What is the smallest number (in decimal) that can be coded with this format?

Sol:
For 9-bit exponents, the bias is $(2^{9-1}) - 1 = 255$.

(a) $(121)_{10} = (1111001)_2 \Rightarrow (121/16)_{10} = (111.1001)_2 = (1.\underline{111001})_2 \times (2^2)_{10}$
$\Rightarrow$ The significand is $\underline{111001}$.
$\Rightarrow$ The exponent $(2)_{10}$ should be codes in biased representation, i.e., $(2 + 255)_{10} = (100000001)_2$.
$\Rightarrow$

sign	exponent	significand
1	100000001	111001

(b)

sign	exponent	significand
1	100001100	111100

$\Rightarrow (100001100)_2 = (268)_{10} \Rightarrow$ The exponent in decimal is $268 - 255 = 13$
$\Rightarrow$ The decimal value is $-(1.111100)_2 \times (2^{13})_{10} = -(2-2^{-4})_{10} \times (2^{13})_{10} = -(2^{14}-2^{9})_{10} = (-15872)_{10}$

(c) The smallest significand possible is $(1)_{10}$, the smallest exponent possible is $(-255)_{10}$, the smallest positive decimal number in this format is 1×2^{-255}.

(d) The largest significand possible is $(2 - 2^{-6})_{10}$, the largest exponent possible is $(256)_{10}$, the smallest decimal number in this format is $-(2 - 2^{-6}) \times 2^{256}$.