

## 제 5 장

5.1  $T_{avg.m.acc} = 20 \times 0.9 + 200 \times (1-0.9) = 18 + 20 = 38 \text{ ns}$

5.2  $T(0) = 10 \times 0 + 100 \times (1-0) = 100 \text{ ns}$

$T(20\%) = 10 \times 0.2 + 100 \times (1-0.2) = 2 + 80 = 82 \text{ ns}$

$T(40\%) = 10 \times 0.4 + 100 \times (1-0.4) = 4 + 60 = 64 \text{ ns}$

$T(60\%) = 10 \times 0.6 + 100 \times (1-0.6) = 6 + 40 = 46 \text{ ns}$

$T(80\%) = 10 \times 0.8 + 100 \times (1-0.8) = 8 + 20 = 28 \text{ ns}$

$T(100\%) = 10 \times 0.0 + 100 \times (1-0.0) = 100 \text{ ns}$

5.3

(1) 000H ~ 1FFH

(2) 000H ~ 3FFH

(3) 000H ~ FFFH

5.4

(1) 32개의 병렬연결과 2개의 직렬연결이 필요하므로  $32 \times 2 = 64$  개

(2) 8개의 병렬연결과 8개의 직렬연결이 필요하므로  $8 \times 8 = 64$  개

(3) 4개의 병렬연결과 16개의 직렬연결이 필요하므로  $4 \times 16 = 64$  개

5.5 64M x 32

(1)  $16 \times 32 = 512$ 개

(2)  $64 \times 8 = 512$ 개

5.6

00000 ~ 11111

5.7

000H ~ 7FFH

5.8

000000 ~ 111111

5.9

0000H ~ 1FFFH

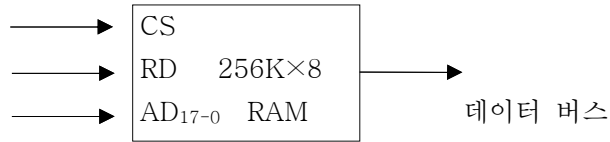
5.10

00000 ~ 11111

5.11

000H ~ 7FFH

5.12



5.13

기억장치 칩	주소영역 (16 진수)	주소 비트들															
		A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
ROM1	0000H ~ 0FFFH	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x
ROM2	1000H ~ 1FFFH	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x
RAM1	8000H ~ 87FFH	1	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x
RAM2	8800H ~ 17FFH	1	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x

5.14

(1) RAM은 2개의 병렬연결과 2개의 직렬연결이 필요하므로  $2 \times 2 = 4$ 개

ROM은 2개의 직렬연결

(2) RAM :  $2\text{Mbyte} \div 2 = 1\text{Mword}$

ROM :  $1\text{Mbyte} \div 2 = 0.5\text{Mword}$

전체 기억장치 용량 : 1.5Mword

(3)

기억장치 칩	주소영역 (16진수)
RAM 1 & 2	000000H ~ 0FFFFFFH
RAM 3 & 4	100000H ~ 1FFFFFFH
ROM 1	200000H ~ 27FFFFFFH
ROM 2	280000H ~ 2FFFFFFH

5.15

$$(1) \frac{7500}{15000} = 0.5 \text{ (50\%)} \quad (2) \frac{12000}{15000} = 0.8 \text{ (80\%)} \quad (3) \frac{13500}{15000} = 0.9 \text{ (90\%)}$$

$$5.16 \quad (0.85 \times 8\text{ns}) + (0.15 \times 100\text{ns}) = 21.8\text{ns}$$

$$5.17 \quad T_a = \{H \times T_c\} + \{(1 - H) \times (T_m + T_k)\}$$

$T_k = 2\text{ns}$  인 경우

$$T_a = (0.85 \times 8\text{ns}) + (0.15 \times 102) = 22.1\text{ns}$$

$$5.18 \quad 19\text{ns} \geq H \times 10\text{ns} + (1 - H) \times 100\text{ns}$$

$$\begin{aligned}
19 &\geq 10H + 100 - 100H \\
90H &\geq 100 - 19 \\
90H &\geq 81 \\
H &\geq 0.9
\end{aligned}$$

5.19

$$(1) \frac{16K}{4} = 4K = 4096\text{개}$$

$$(2) \frac{16K}{8} = 2K = 2048\text{개}$$

$$(3) \frac{16K}{16} = 1K = 1024\text{개}$$

5.20

$$(1) \frac{16K}{4} = 4K = 4096\text{개}$$

$$(2) \frac{1M}{4} = 256K = 2^{18}\text{개}$$

$$(3) 2^{18} / 2^{12} = 2^6 = 64\text{개}$$

(4)

태그(tag)	라인(slot)	단어
6	12	2

5.21

태그(tag)	세트(set)	단어
8	4	4

5.22

·직접 사상 방식

- 장점 : 구현하는 비용이 적게 든다.
- 단점 : 각 주기억장치 블록이 적재될 수 있는 캐시 라인이 한 개 뿐이기 때문에, 프로그램의 수행과정에서 같은 라인에 사상되는 두 개의 블록들로부터 데이터들을 번갈아 읽어 와야 한다. 따라서 그 블록들은 캐시에서 반복적으로 교체되어, 적중률이 낮아진다.

·완전-연관 사상 방식

- 장점 : 주기억장치 블록이 캐시의 어떤 라인으로든 적재될 수 있도록 허용함으로써 순차적인 프로그램 세그먼트나 데이터 배열 전체가 캐시로 적재될 수 있으므로 지역성이 높다면 적중률이 매우 높아질 것이다.
- 완전-연관 사상의 단점 : 캐시 라인들의 태그들을 병렬로 검사하기 위하여 매우 복잡하고 비용이 높은 회로를 포함해야 한다는 결정적인 단점이 있기 때문에 실제 시스템에서는 거의 사용되지 않고 있다.

5.23

- (1) 캐시 적중
- (2) 캐시 미스, 결과 = 11 'elec'
- (3) 캐시 미스, 결과 = 00 'assm'
- (4) 캐시 적중

5.24

- (1) 캐시 미스, 결과 = 110 'comp'
- (2) 캐시 미스, 결과 = 011 'move'
- (3) 캐시 적중
- (4) 캐시 미스, 결과 = 000 'hand'

5.25

(라인의 수가 2개인 경우),  $H = \frac{2}{11}$

(라인의 수가 4개인 경우),  $H = \frac{6}{11}$

5.26

- (1) FIFO, 라인 수 2개 =  $3/12 = 1/4$
- (2) FIFO, 라인 수 4개 =  $5/12$
- (3) LRU 알고리즘, 라인 수 2개 =  $5/12$
- (4) LRU 알고리즘, 라인 수 4개 =  $6/12$

5.27

- (1) Write-through

읽기평균 :  $0.8 \times 20\text{ns} + 0.2 \times 200\text{ns} = 16\text{ns} + 40\text{ns} = 56\text{ns}$

쓰기평균 :  $200\text{ns}$

평균 기억장치 액세스 시간 :  $0.6 \times 56\text{ns} + 0.4 \times 200\text{ns} = 113.6\text{ns}$

- (2) Write-back

$0.8 \times 20\text{ns} + 0.2 \times (200\text{ns} + 0.3 \times 200\text{ns}) = 16 + 52 = 68\text{ns}$

5.28

식 5-6 :  $T_a = H_1 \times T_{L1} + (H_2 - H_1) \times T_{L2} + (1 - H_2) \times T_m$

$T_a = 0.6 \times 2\text{ns} + (0.85 - 0.6) \times 20\text{ns} + (1 - 0.85) \times 200\text{ns}$   
 $= 1.2 + 5 + 30 = 36.2\text{ns}$

5.29

식 5-7 :  $T_a = H_1 \times T_{L1} + (1 - H_1)H_2 \times T_{L2} + \{1 - H_1 - (1 - H_1)H_2\} \times T_m$

$T_a = 0.6 \times 2\text{ns} + (0.4) \times 0.85 \times 20\text{ns} + \{1 - 0.6 - (0.4 \times 0.85)\} \times 200\text{ns}$

$$= 1.2 + 6.8 + 12 = 20\text{ns}$$

5.30

DDR SDRAM : 데이터 폭 = 64bits = 8bytes

대역폭(초당 데이터 전송량) = 버스 폭  $\times$  클록 주파수

- (1)  $8\text{bytes} \times 100\text{MHz} \times 2 = 1600 \text{ Mbytes/sec}$
- (2) 2656 Mbytes/sec
- (3) 3200 Mbytes/sec

5.31

- (1)  $8\text{bytes} \times 133\text{MHz} \times 4 = 4256 \text{ Mbytes/sec}$
- (2) 6400 Mbytes/sec
- (3) 8512 Mbytes/sec

5.32

$$5\text{ns} \times 4\text{사이클} + (2 \times 5\text{ns}) = 30\text{ns}$$

5.33

1Gx4 경우,

- (a) 구성 불가 (64비트 데이터 입출력 불가능)
- (b) 구성 불가 (각 면당 64비트 데이터 입출력 불가능)
- (c)  $(1\text{Gx4}) \times 16\text{개} = 8 \text{ GBytes}$

512Mx8 경우,

- (a)  $(512\text{M} \times 8) \times 8\text{개} = 4 \text{ GBytes}$
- (b)  $\{(512\text{M} \times 8) \times 8\text{개}\} \times 2\text{면} = 8 \text{ GBytes}$
- (c)  $(512\text{M} \times 8) \times 16\text{개} = 8 \text{ GBytes}$