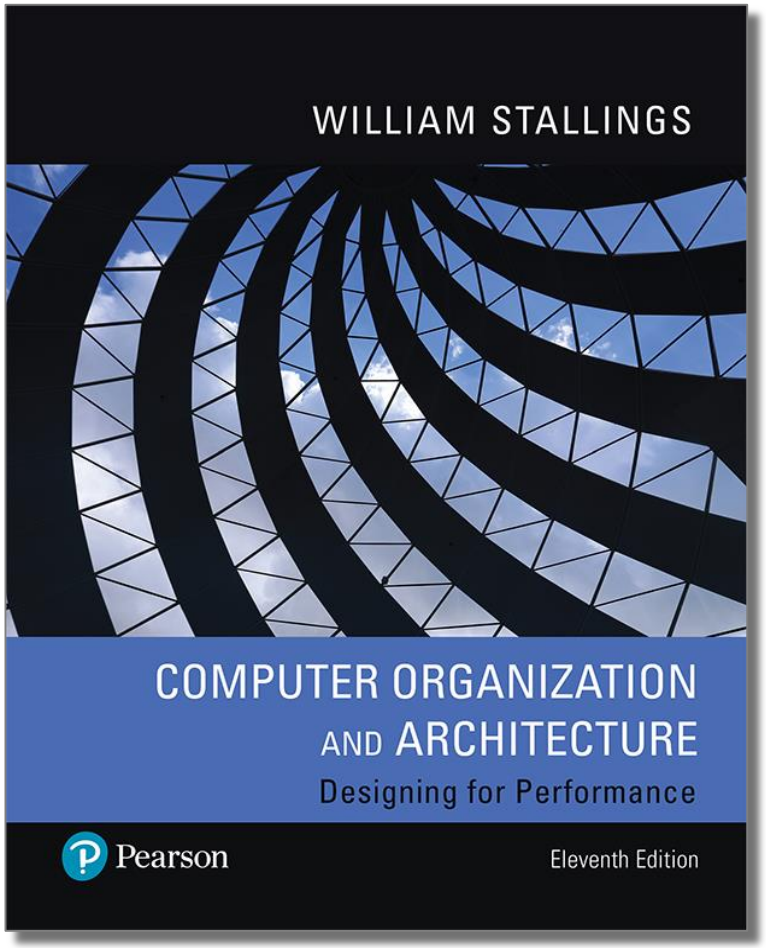
Computer Organization and Architecture

Designing for Performance

11th Edition



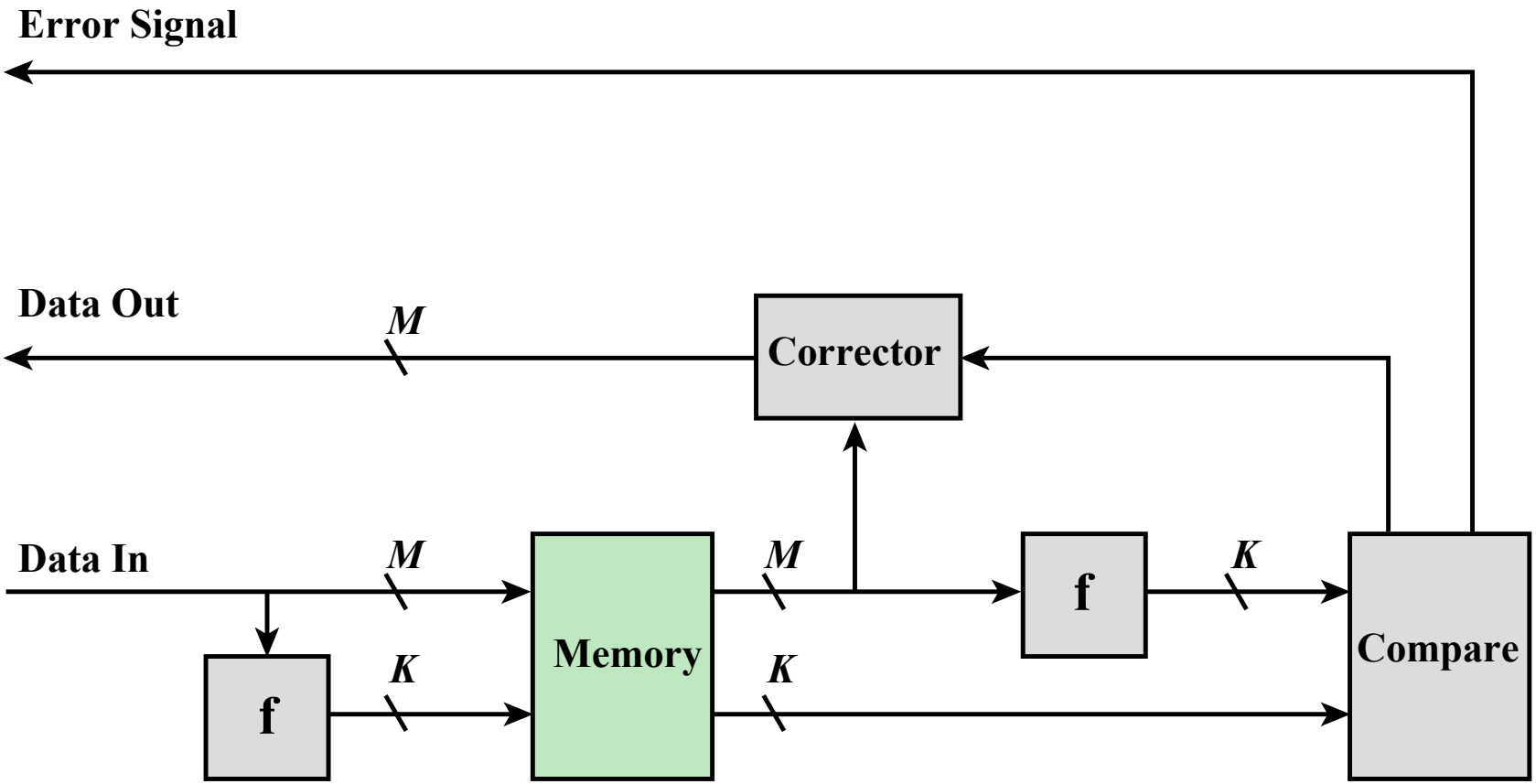
Chapter 6

Internal Memory

Error Correction

- Hard Failure
  - Permanent physical defect
  - Memory cell or cells affected cannot reliably store data but become stuck at 0 or 1 or switch erratically between 0 and 1
  - Can be caused by:
    - Harsh environmental abuse
    - Manufacturing defects
    - Wear
- Soft Error
  - Random, non-destructive event that alters the contents of one or more memory cells
  - No permanent damage to memory
  - Can be caused by:
    - Power supply problems
    - Alpha particles

Figure 6.7
Error-Correcting Code Function



Error Correction

- One way is to have redundant copies of the data to check if there is an error in one of the copies. For example:
  - 1 0 1 **1** 0
  - 1 0 1 **1** 0
  - 1 0 1 **0** 0
  - You can see that the first two of the copies are “1”s in the same position but the third is a “0”. This means, the third one needs to be corrected by flipping it to a “1”.
- This approach is costly in terms of additional space!

- The goal is to do error correction while reducing the additional storage required as much as possible.

Figure 6.8
Hamming Error-Correcting Code

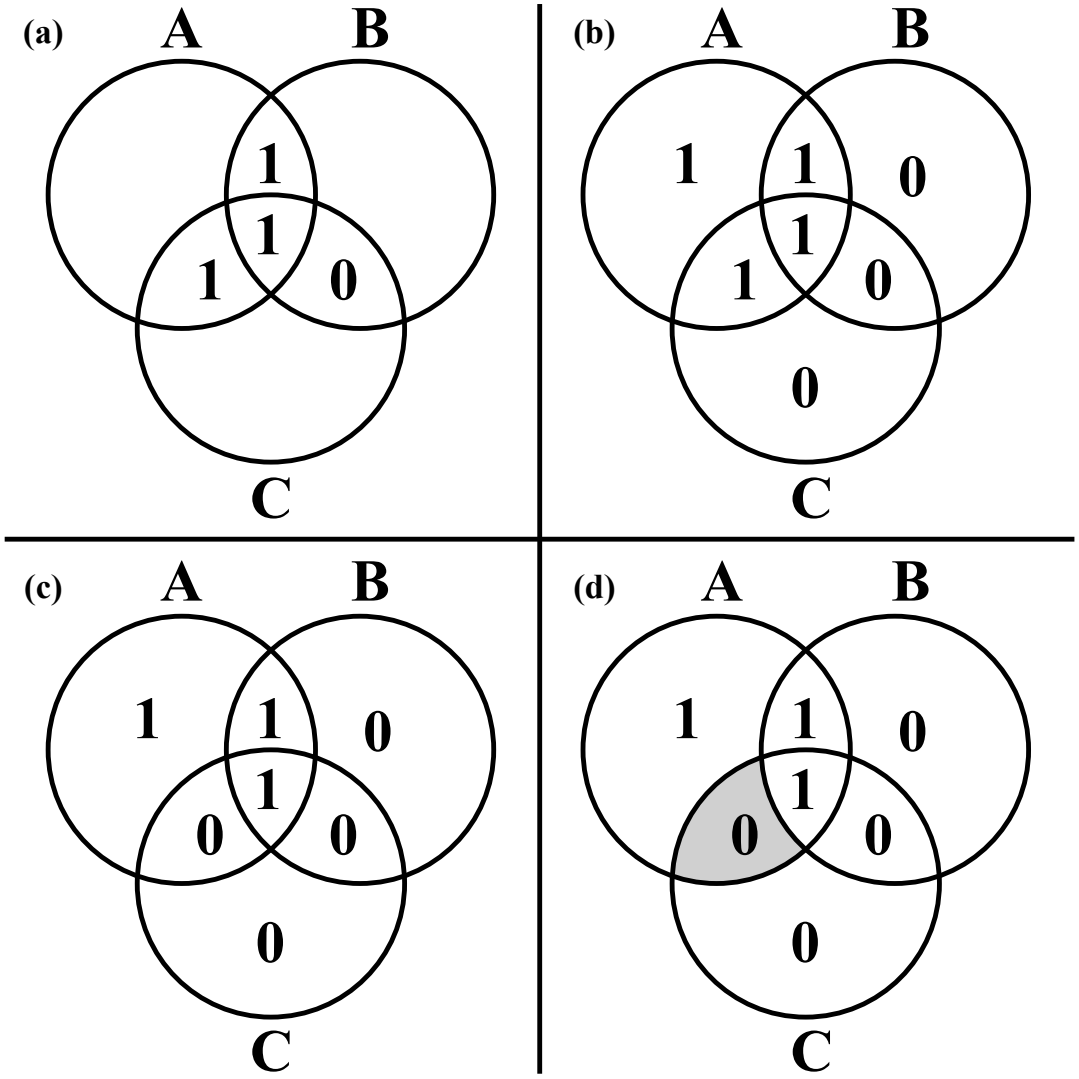


Table 6.2
Increase in Word Length with Error Correction

	Single-Error Correction		Single-Error Correction/ Double-Error Detection	
Data Bits	**Check Bits**	**% Increase**	**Check Bits**	**% Increase**
8	4	50.0	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

Figure 6.9
Layout of Data Bits and Check Bits

- Assume that you have a 12-bit space. How many will be reserved for the check bits and data bits?

Figure 6.9
Layout of Data Bits and Check Bits

- Assume that you have a 12-bit space. How many will be reserved for the check bits and data bits?

Bit Position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data Bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C8				C4		C2	C1

Power of 2s positions

How to Calculate Check Bits

- Each check bit operates on every data bit whose position number contains a 1 in the same position as the position number of that check bit.
- For example, D1 (pos. 3), D2 (pos. 5), D4 (pos. 7), D5 (pos. 9) and D7 (pos. 11) all contain 1 in the least significant bit as does C1.

Hamming Code Example

- Assume that the 8-bit input word is 00111001 with data bit D1 in the rightmost position. Compute the check bits.
- To calculate check bit “C1”:

12	11	10	9	8	7	6	5	4	3	2	1
1100	101**1**	1010	100**1**	1000	011**1**	0110	010**1**	0100	001**1**	0010	000**1**
D8	**D7**	D6	**D5**	C8	**D4**	D3	**D2**	C4	**D1**	C2	**C1**
0	0	1	1		1	0	0		1		**1**

$$C1 = D1 + D2 + D4 + D5 + D7$$
$$= 1 + 0 + 1 + 1 + 0$$
$$= 1$$

Hamming Code Example

- Assume that the 8-bit input word is 00111001 with data bit D1 in the rightmost position. Compute the check bits.
- To calculate check bit “C2”:

12	11	10	9	8	7	6	5	4	3	2	1
1100	10**1**1	10**1**0	1001	1000	01**1**1	01**1**0	0101	0100	00**1**1	00**1**0	0001
D8	**D7**	**D6**	D5	C8	**D4**	**D3**	D2	C4	**D1**	**C2**	C1
0	0	1	1		1	0	0		1	**1**	1

$$C2 = D1 + D3 + D4 + D6 + D7$$
$$= 1 + 0 + 1 + 1 + 0$$
$$= 1$$

Hamming Code Example

- Assume that the 8-bit input word is 00111001 with data bit D1 in the rightmost position. Compute the check bits.
- To calculate check bit “C4”:

12	11	10	9	8	7	6	5	4	3	2	1
1**1**00	1011	1010	1001	1000	0**1**11	0**1**10	0**1**01	0**1**00	0011	0010	0001
D8	D7	D6	D5	C8	**D4**	**D3**	**D2**	**C4**	D1	C2	C1
0	0	1	1		1	0	0	**1**	1	1	1

$$C4 = D2 + D3 + D4 + D8$$
$$= 0 + 0 + 1 + 0$$
$$= 1$$

Hamming Code Example

- Assume that the 8-bit input word is 00111001 with data bit D1 in the rightmost position. Compute the check bits.
- To calculate check bit “C8”:

12	11	10	9	8	7	6	5	4	3	2	1
1100	**1**011	**1**010	**1**001	**1**000	0111	0110	0101	0100	0011	0010	0001
D8	**D7**	**D6**	**D5**	**C8**	D4	D3	D2	C4	D1	C2	C1
0	0	1	1	**0**	1	0	0	1	1	1	1

$$C8 = D5 + D6 + D7 + D8$$
$$= 1 + 1 + 0 + 0$$
$$= 0$$

Hamming Code Example

- Assume that the 8-bit input word is 00111001 with data bit D1 in the rightmost position. Compute the check bits.
- To calculate check bit “C8”:

12	11	10	9	8	7	6	5	4	3	2	1
1100	**1**011	**1**010	**1**001	**1**000	0111	0110	0101	0100	0011	0010	0001
D8	**D7**	**D6**	**D5**	**C8**	D4	D3	D2	C4	D1	C2	C1
0	0	1	1	**0**	1	0	0	1	1	1	1

$$C8 = D5 + D6 + D7 + D8$$
$$= 1 + 1 + 0 + 0$$
$$= 0$$

Check bits: 0111

Hamming Code Example

- Now suppose that the data bit 3 sustains an error and is change from 0 to 1

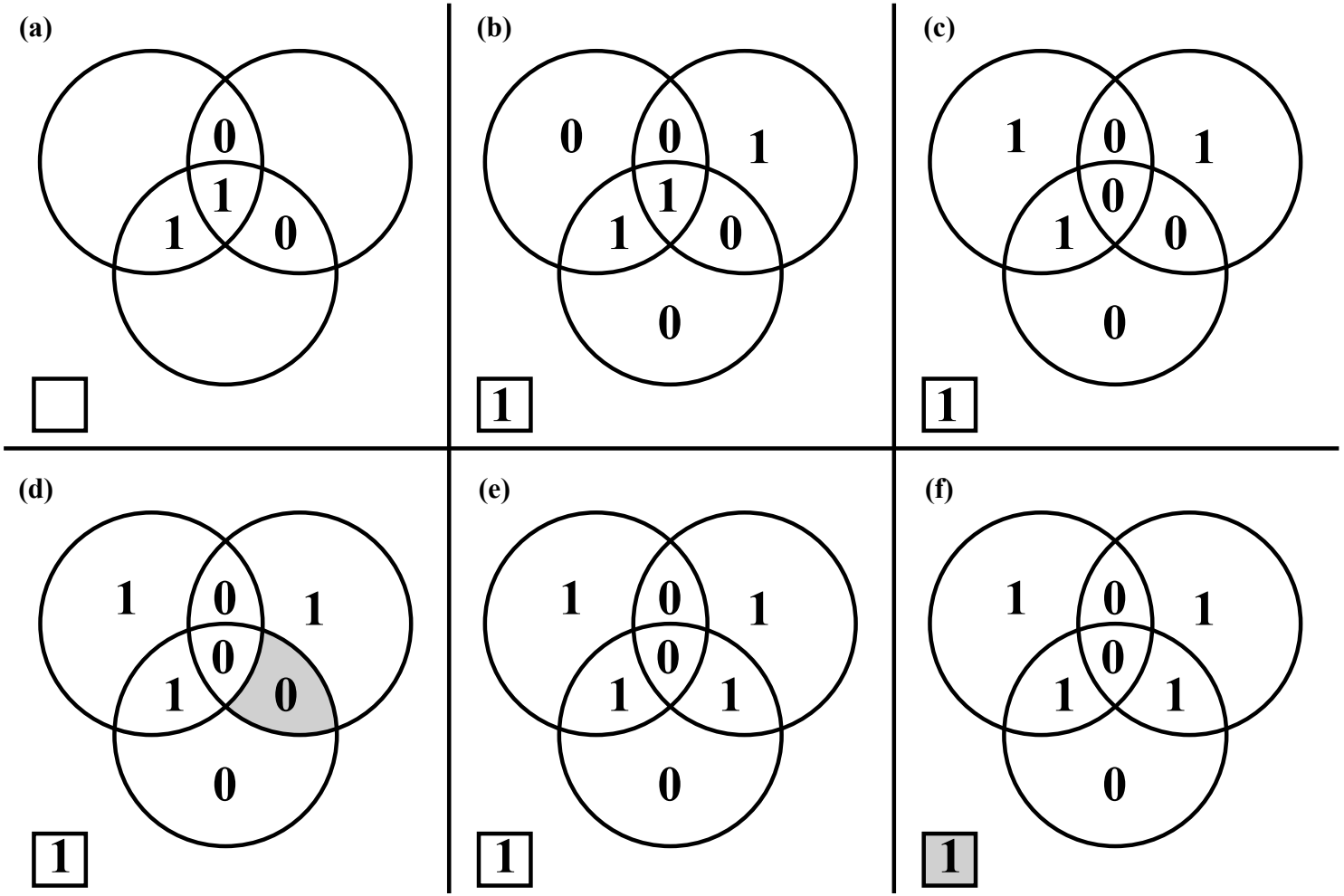
12	11	10	9	8	7	6	5	4	3	2	1
1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
D8	D7	D6	D5	C8	D4	D3	D2	C4	D1	C2	C1
0	0	1	1	0	1	1	0	1	1	1	1

- When you try to fetch the word, you will calculate the check bits again from the data bits.
  - You will get:**0001**
  - You compare it to the original stored check bits: **0111 + 0001 = 0110**
  - It means you have an error in the bit at **position 6**
  - You correct the error by flipping this bit.

Figure 6.10
Check Bit Calculation

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1
Word stored as	0	0	1	1	0	1	0	0	1	1	1	1
Word fetched as	0	0	1	1	0	1	1	0	1	1	1	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Check Bit					0				0		0	1

Figure 6.11
Hamming SEC-DEC Code



Advanced DRAM Organization

- One of the most critical system bottlenecks when using high-performance processors is the interface to main internal memory
- The traditional DRAM chip is constrained both by its internal architecture and by its interface to the processor's memory bus
- A number of enhancements to the basic DRAM architecture have been explored
  - The schemes that currently dominate the market are SDRAM and DDR-DRAM

SDRAM

DDR-DRAM

RDRAM

Synchronous DRAM (SDRAM)

One of the most widely used forms of DRAM

Exchanges data with the processor synchronized to an external clock signal and running at the full speed of the processor/memory bus without imposing wait states

With synchronous access the DRAM moves data in and out under control of the system clock

- The processor or other master issues the instruction and address information which is latched by the DRAM
- The DRAM then responds after a set number of clock cycles
- Meanwhile the master can safely do other tasks while the SDRAM is processing

Double Data Rate SDRAM (DDR SDRAM)

- Developed by the JEDEC Solid State Technology Association (Electronic Industries Alliance's semiconductor-engineering-standardization body)
- Numerous companies make DDR chips, which are widely used in desktop computers and servers
- DDR achieves higher data rates in three ways:
  - First, the data transfer is synchronized to both the rising and falling edge of the clock, rather than just the rising edge
  - Second, DDR uses higher clock rate on the bus to increase the transfer rate
  - Third, a buffering scheme is used

Table 6.4
DDR Characteristics

	DDR1	DDR2	DDR3	DDR4
Prefetch buffer (bits)	2	4	8	8
Voltage level (V)	2.5	1.8	1.5	1.2
Front side bus data rates (Mbps)	200—400	400—1066	800—2133	2133—4266

Copyright