Digital design Hwk1

Tuesday, October 1, 2019 12:49 PM

1.3 Assume that 0 V is encoded as 00, 1 V as 01, 2 V as 10, and 3 V as 11. You are given a digital encoding of an audio signal as follows: 1111101001010000. Plot the re-created analog signal with time on the *x*-axis and voltage on the *y*-axis. Assume that each encoding's corresponding voltage should be output for 1 millisecond.

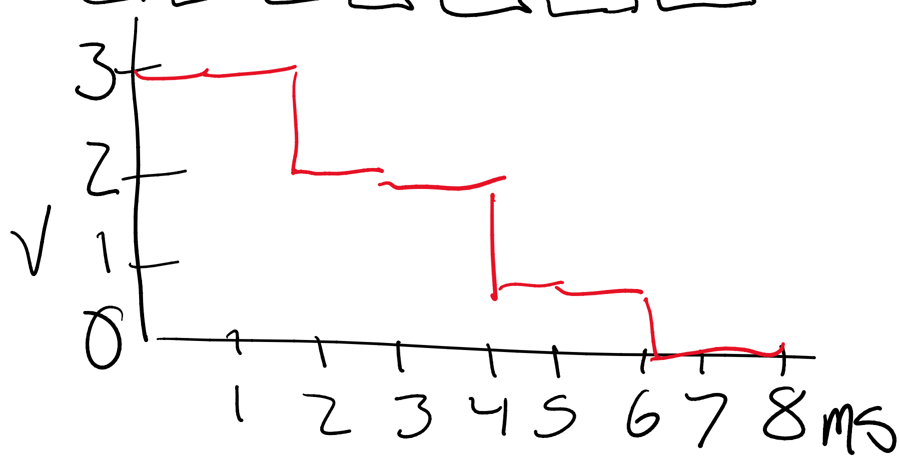
0V -> 00

1V -> 01

2V -> 10

3V -> 11

11 11 10 10 01 01 00 00



1.4 Assume that a signal is encoded using 12 bits. Assume that many of the encodings turn out to be either 000000000000, 000000000001, or 111111111111. We thus decide to create compressed encodings by representing 000000000000 as 00, 000000000001 as 01, and 111111111111 as 10. 11 means that an uncompressed encoding follows. Using this encoding scheme, decompress the following encoded stream:

00 00 01 10 11 010101010101 00 00 10 10

0000 0000 0000 -> 00

0000 00000001 -> 01

1111 1111 1111 -> 10

11 ?

00 00 01 10 11 010101010101 00 00 10 10

1 2 3 4 5 6 7 8 9 10

0000 0000 0000 (1) 0000 0000 0000 (2)

0000 0000 0001 (3) 1111 1111 1111 (4)

0101 0101 0101 (5&6) 0000 0000 0000 (7)

0000 0000 0000 1111 1111 1111 11111111 1111

5&6

0000 0000 0000 1111 1111 1111 11111111 1111

8 9 10

1.10 Convert the following binary numbers to decimal numbers:

(a) 000011
(b) 1111
(c) 11110
(d) 111100
(e) 0011010

64 32 16 8 4 2 1

(a) 000011
(b) 1111
(c) 11110
(d) 111100
(e) 0011010

(a) $\cancel{0 \cdot 32} + \cancel{0 \cdot 16} + \cancel{0 \cdot 8} + \cancel{0 \cdot 4} + 1 \cdot 2 + 1 \cdot 1$

$1 \cdot 2 + 1 \cdot 1 = \boxed{3}$

(b) $1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1$

$8 + 4 + 2 + 1 = \boxed{15}$

(c) $1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$

$16 + 8 + 4 + 2 + 0 = \boxed{30}$

(d) $1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1$

$32 + 16 + 8 + 4 + 0 + 0 = \boxed{60}$

(e) $0 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$

$0 + 0 + 16 + 8 + 0 + 2 + 0 = \boxed{26}$

1.13 Convert the following decimal numbers to binary numbers using the addition method:

(a) 3
(b) 65
(c) 90
(d) 100

(a) 3

(b) 65

(c) 90

(d) 100

(a) 3

0	1	1
4	2	1

$= \boxed{11}$

(b) 65

0	1	0	0	0	0	0	1
128	64	32	16	8	4	2	1

$= \boxed{1000001}$

(c) 90

0	1	0	1	1	0	1	0
128	64	32	16	8	4	2	1

$= \boxed{1011010}$

(d) 100

0	1	1	0	0	1	0	0
128	64	32	16	8	4	2	1

$= \boxed{1100100}$

1.15 Convert the following decimal numbers to binary numbers using the divide-by-2 method:
(a) 19
(b) 30
(c) 64
(d) 128

(a) 19

(b) 30

(c) 64

(a) $2 \overline{)19}$ = 9, remainder 1

$2 \overline{)9}$ = 4, remainder 1

$2 \overline{)2}$ = 1, remainder 0

$2 \overline{)1}$ = 0, remainder 1

$\boxed{10011}$

(c) 6 .

(d) 128

1

$2\overline{)4}$ = 2, remainder 0

T

(b) $2\overline{)30}$ = 15, remainder 0

$2\overline{)15}$ = 7, remainder 1

$2\overline{)7}$ = 3, remainder 1

$2\overline{)3}$ = 1, remainder 1

$2\overline{)1}$ remainder 1

$\boxed{11110}$

(c) $2\overline{)64}$ = 32, remainder 0

$2\overline{)32}$ = 16, remainder 0

$2\overline{)16}$ = 8, remainder 0

$2\overline{)8}$ = 4, remainder 0

$2\overline{)4}$ = 2, remainder 0

$2\overline{)2}$ = 1, remainder 0

$2\overline{)1}$ remainder 1

$\boxed{1000000}$

(d) $2\overline{)128}$ 64, 0

$2\overline{)64}$ 32, 0

$2\overline{)32}$ 16, 0

$2\overline{)16}$ 8, 0

$2\overline{)8}$ 4, 0

$2\overline{)4}$ 2, 0

$2\overline{)2}$ 1, 0

$2\overline{)1}$ 1

$\boxed{10000000}$

1.17 Convert the following decimal numbers to binary numbers using the divide-by-2 method:
(a) 23
(b) 87
(c) 123
(d) 101

(a) 23
(b) 87
(c) 123
(d) 101

(a) $2\overline{)23}$ 11, 1

$2\overline{)11}$ 5, 1

$2\overline{)5}$ 2, 1

$2\overline{)2}$ 1, 0

$2\overline{)1}$ 0, 1

$\boxed{10111}$

(b) $2\overline{)87}$ 43, 1

$2\overline{)10}$ 5

$\boxed{1010111}$

$2\overline{)87}$ 1

$2\overline{)10}$ 0

$\boxed{1010111}$

$2\overline{)43}$ quotient 21, remainder 1

$2\overline{)5}$ quotient 2, remainder 1

$2\overline{)1}$ quotient 0, remainder 1

$2\overline{)21}$ quotient 10, remainder 1

$2\overline{)2}$ quotient 1, remainder 0

(c)

$2\overline{)123}$ quotient 61, remainder 1

$2\overline{)15}$ quotient 7, remainder 1

$\boxed{1111011}$

$2\overline{)61}$ quotient 30, remainder 1

$2\overline{)7}$ quotient 3, remainder 1

$2\overline{)1}$ quotient 0, remainder 1

$2\overline{)30}$ quotient 15, remainder 0

$2\overline{)3}$ quotient 1, remainder 1

(d)

$2\overline{)101}$ quotient 50, remainder 1

$2\overline{)12}$ quotient 6, remainder 0

$\boxed{1100101}$

$2\overline{)1}$ quotient 0, remainder 1

$2\overline{)50}$ quotient 25, remainder 0

$2\overline{)6}$ quotient 3, remainder 0

$2\overline{)...}$ quotient 12

$2\overline{)3}$ quotient 1

$$2\overline{)25}^{\,12} \quad 1$$

$$2\overline{)3}^{\,1} \quad 1$$

1.19 Convert the following binary numbers to hexadecimal:
(a) 11001101
(b) 10100101
(c) 11110001
(d) 1101101111100

(a) 11001101
(b) 10100101
(c) 11110001
(d) 1101101111100

(a) $\underbrace{1100}_{C}\ \underbrace{1101}_{D}$

$\boxed{CD}$

(b) $\underbrace{1010}_{A}\ \underbrace{0101}_{5}$

$\boxed{A5}$

(c) $\underbrace{1111}_{F}\ \underbrace{0001}_{1}$

$\boxed{F1}$

(d) $\underbrace{1}_{1}\ \underbrace{1011}_{B}\ \underbrace{0111}_{7}\ \underbrace{1100}_{C}$

$\boxed{1B7C}$

$\boxed{F1}$

1.23 Convert the following hexadecimal numbers to binary:
(a) B0C4
(b) 1EF03
(c) F002
(d) BEEF

(a) B0C4
(b) 1EF03
(c) F002
(d) BEEF

(a) $\underbrace{1011}_{B}\ \underbrace{0000}_{0}\ \underbrace{1100}_{C}$

$\underbrace{0100}_{4}$

$\boxed{1011000011000100}$

(b) 1EF03

$\underbrace{0001}_{1}\ \underbrace{1110}_{E}\ \underbrace{1111}_{F}\ \underbrace{0000}_{0}\ \underbrace{0011}_{3}$

$\boxed{11110111100000011}$

(c) F002

$\underbrace{1111}_{F}\ \underbrace{0000}_{0}\ \underbrace{0000}_{0}\ \underbrace{0010}_{2}$

$\boxed{1111000000000010}$

(d) BEEF

$\underbrace{1011}_{B}\ \underbrace{1110}_{E}\ \underbrace{1110}_{E}\ \underbrace{1111}_{F}$

$\boxed{1011111011101111}$

1.25 Convert the following hexadecimal numbers to decimal:

(a) 10
(b) 4E3
(c) FF0
(d) 200

(a) 10 (a) 00010000

(a) 10

(b) 4E3

(c) FF0

(d) 200

(a) 00010000

16 8 4 2 1

$\boxed{16}$

(b) 0100 1110 0011

1 5 2 128 64 32 16 8 4 2 1
0 1 5
2 2 6
4

1024 + 128 + 64 + 32 + 2 + 1

$\boxed{1251}$

(c) 1 1 1 1 1 1 1 1 0 0 0 0

2 1 5 2 128 64 32 16 8 4 2 1
0 0 1 5
4 2 2 6
8 4

2048 + 1024 + 512 + 256 + 128 + 64 + 32 + 16

$\boxed{4080}$

(d) 0010 0000 0000

$\boxed{512}$

(d) 200

1.26 Convert the decimal number 128 to the following number systems:
(a) binary
(b) hexadecimal
(c) base three
(d) base five
(e) base fifteen

128 (a) $\underbrace{1000}_{8}\ \underbrace{0000}_{0}$

(b) 80

(c) $128 = 42 \cdot 3 + 2$

$42 = 14 \cdot 3 + 0$

$14 = 4 \cdot 3 + 2$

$4 = 1 \cdot 3 + 1$

$1 = 0 \cdot 3 + 1$

$\boxed{11202}$

(d) $128 = 25 \cdot 5 + 3$

$25 = 5 \cdot 5 + 0$

$5 = 1 \cdot 5 + 0$

$1 = 0 \cdot 5 + 1$

$\boxed{1003}$

(e) $128 = 8 \cdot 15 + 8$

$8 = 0 \cdot 15 + 8$

$\boxed{88}$

1.27 Compare the number of digits necessary to represent the following decimal numbers in binary, octal, decimal, and hexadecimal representations. You need not determine the actual representations—just the number of required digits. For example, representing the decimal number 12 requires four digits in binary (1100 is the actual representation), two digits in octal (14), two digits in decimal (12), and one digit in hexadecimal (C).

(a) 8
(b) 60
(c) 300
(d) 1000
(e) 999,999

ⓐ 4 in binary (1000), 2 in octal (10),
1 in decimal (8), 1 in hex (8)

ⓑ 6 in binary (111100), 2 in octal (74),
2 in decimal (60), 2 in hex (3C)

ⓒ 9 in binary (100101100), 3 in octal (454),
3 in decimal (300), 3 in hex (12C)

ⓓ 10 in binary (1111101000), 4 in octal (1750),
4 in decimal (1000), 3 in hex (3E8)

ⓔ 20 in binary (11110100001000111111),
7 in octal (3641077),
6 in decimal (999999),
5 in hex (F423F)

1.30 Use a microprocessor like that in Figure 1.23 to implement a system that sounds an alarm whenever there is motion detected at the same time in three different rooms. Each room's motion sensor is an input to the microprocessor; a 1 means motion, a 0 means no motion. The microprocessor can sound the alarm by setting an output wire "alarm" to 1. Show the connections to and from the microprocessor, and the C code to execute on the microprocessor. Hint: this problem is similar to Example 1.10.

motion sensor 1

alarm

motion sensor 1 —

" " 2

" " 3

alarm

all have to be "1" at same time, so

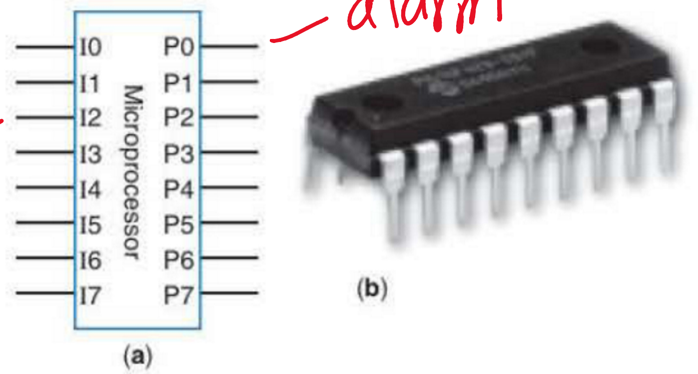


Figure 1.23 A basic microprocessor: (a) with eight outputs *P0-P7* that each can be set to a 0 or 1, and eight inputs *I0-I7* too, (b) photograph of a real microprocessor package.

```
while(1){
   P0 = I0 && I1 && I2;
}
```

1.32 Suppose that a particular banking system supports encrypted transactions, and that decrypting each transaction consists of three sub-tasks A, B, and C. The execution times of each task on a microprocessor versus a custom digital circuit are 50 ms versus 1 ms for A, 20 ms versus 2 ms for B, and 20 ms versus 1 ms for C. Partition the tasks among the microprocessor and custom digital circuitry, such that you minimize the amount of custom digital circuitry, while meeting the constraint of decrypting at least 40 transactions per second. Assume that each task requires the same amount of digital circuitry.

1.33 How many possible partitionings are there of a set of *N* tasks, where each task can be imple-

40 per 1s   1/40 = 0.025 s/trasaction

25ms

3 micros 50 + 20 + 20 = 90 ms

A(DC) 2 micros 1 + 20 + 20 = 41 ms

✓ $\boxed{\text{A(DC)B(DC)C(micro)}}$ = 1 + 2 + 20 = $\boxed{23\text{ ms}}$

$\boxed{A(DC)B(DC)C(micro)} = 1+2+20 = \boxed{23\,ms}$

sensed).

2.11 Convert the following English problem statements to Boolean equations. Introduce Boolean variables as needed.

(a) A flood detector should turn on a pump if water is detected and the system is set to enabled.

(b) A house energy monitor should sound an alarm if it is night and light is detected inside a room but motion is not detected.

(c) An irrigation system should open the sprinkler's water valve if the system is enabled and neither rain nor freezing temperatures are detected.

a) P = W AND S

b) A = N AND L AND NOT M

c) V = S AND NOT (R OR F)

2.14 Evaluate the Boolean equation F = a AND (b OR (c AND d)) for the given values of variables a, b, c, and d:

(a) a=1, b=1, c=0, d=1
(b) a=0, b=0, c=0, d=1
(c) a=1, b=0, c=0, d=0
(d) a=1, b=0, c=1, d=1

a) F = 1 AND(1 OR(0 AND 1)) = 1

b) F = 0 AND(0 OR(0 AND 1)) = 0

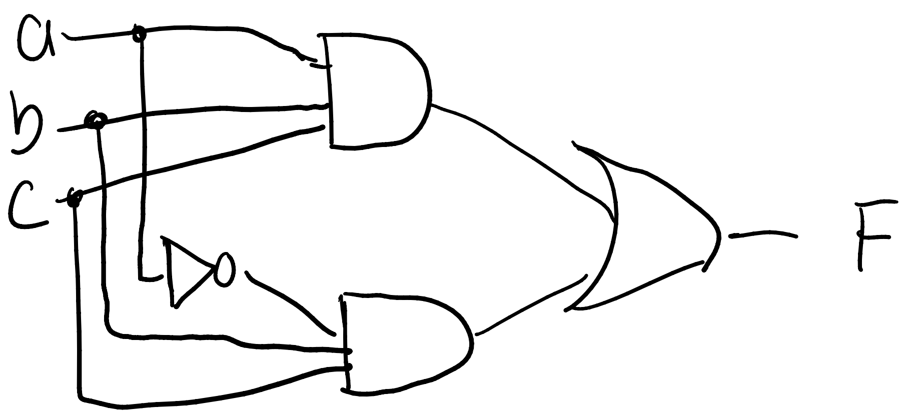
c) F = 1 AND(0 OR(0 AND 0)) = 0

d) F = 1 AND(0 OR(1 AND 1)) = 1

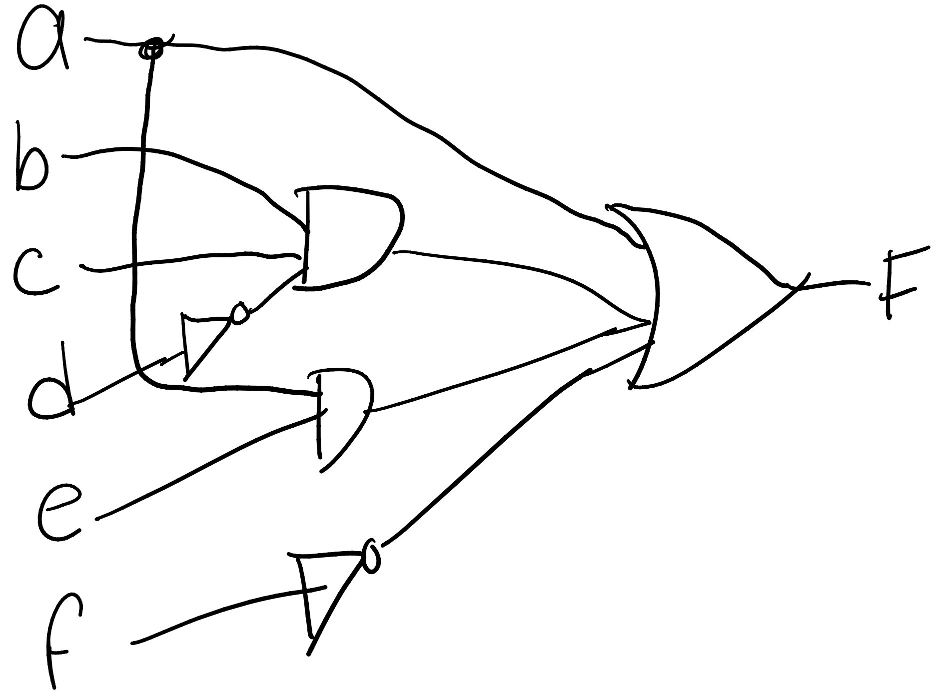
2.19 Convert each of the following equations directly to gate-level circuits:

(a) F = abc + a'bc
(b) F = a + bcd' + ae + f'
(c) F = (a + b) + (c' * (d + e + fg))

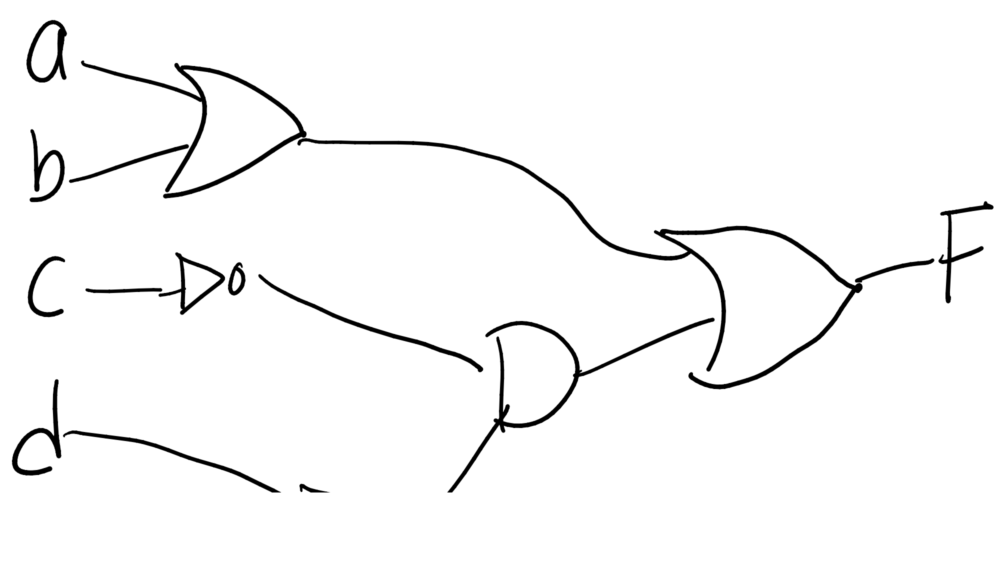
a

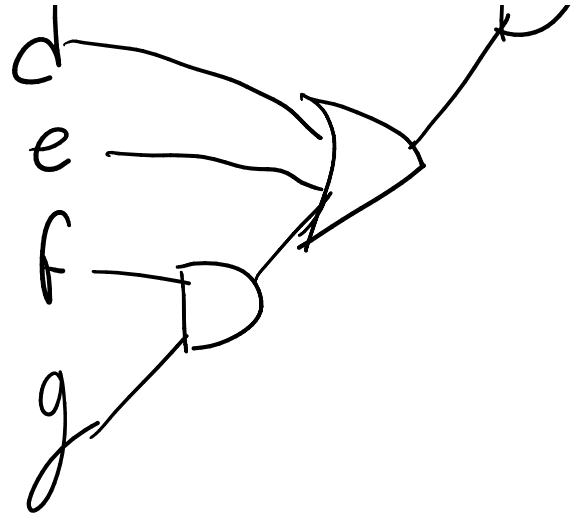


b



c





convert the equation to a circuit using AND, OR, and NOT gates.

2.21 A DJ ("disc jockey," meaning someone who plays music at a party) would like a system to automatically control a strobe light and disco ball in a dance hall depending on whether music is playing and people are dancing. A sound sensor has output S that when 1 indicates that music is playing, and a motion sensor has output M that when 1 indicates that people are dancing. The strobe light has an input L that when 1 turns the light on, and the disco ball has an input B that when 1 turns the ball on. The DJ wants the disco ball to turn on only when music is playing and nobody is dancing, and wants the strobe light to turn on only when music is playing and people are dancing. Create equations describing the desired behavior for B and for L, and then convert each to a circuit using AND, OR, and NOT gates,

S AND NOT M = B   S AND M = L