Home assignment 1

ECSE 104L

**Due date – 19th Feb 2019**

**Late submission not allowed**

**Hand written assignment in a notebook**

1. Convert the following numbers with the indicated bases to decimal:

(a) (4310) 5 (b) (198) 12

(c) (435) 8 (d) (345) 6

2. Determine the base of the numbers in each case for the following operations to be correct:

(a) 14/2 = 5 (b) 54/4 = 13 (c) 24 + 17 = 40.

3. Convert the hexadecimal number 64CD to binary, and then convert it from binary to octal.

4. Convert the decimal number 431 to binary in two ways:

(a) convert directly to binary;

(b) convert first to hexadecimal and then from hexadecimal to binary. Which method is faster?

5. Express the following numbers in decimal:

(a) (10110.0101) 2 (b) (16.5) 16

(c) (26.24) 8 (d) (DADA.B) 16

(e) (1010.1101) 2

6. Obtain the 1’s and 2’s complements of the following binary numbers:

(a) 00010000 (b) 00000000

(c) 11011010 (d) 10101010

(e) 10000101 (f) 11111111.

7. Represent the unsigned decimal numbers 791 and 658 in BCD, and then show the steps

necessary to form their sum.

8. Represent the decimal number 6248 in

(a) BCD, (b) excess‐3 code, (c) 2421 code, (d) 6311 code.

9. Simplify the following Boolean expressions to a minimum number of literals:

(a) *ABC* + *A*’*B* + *ABC*’ (b)*x* ‘ *yz* + *xz*

(c) (*x* + *y*)’ (*x*’ + *y*’) (d)*xy* + *x(wz* + *wz*’*)*

(e) (*BC*’ + *A*’*D*) (*AB*’ + *CD*’) (f) (*a*’ + *c*’) (*a* + *b*’ + *c*’)

10. Simplify the following Boolean functions *T*1 and *T*2 to a minimum number of literals:

*A B C T*1 *T* 2

0 0 0 1 0

0 0 1 1 0

0 1 0 1 0

0 1 1 0 1

1 0 0 0 1

1 0 1 0 1

1 1 0 0 1

1 1 1 0 1

11. Draw the logic diagram corresponding to the following Boolean expressions without simplifying

them:

(a) *BC*’ + *AB* + *ACD*

(b) (*A* + *B*)(*C* + *D*)(*A*’ + *B* + *D*)

(c) (*AB* + *A*’*B*’)(*CD*’ + *C*’*D*)

12. Implement the following Boolean function with NAND gates:

*F* (*x*, *y*, *z*) = ∑(1, 2, 3, 4, 5, 7)

13. Simplify the following Boolean functions, using three-variable k-maps:

(a) *F* (*x*, *y*, *z*) =∑ (0, 1, 5, 7) (b) *F* (*x*, *y*, *z*) = ∑ (1, 2, 3, 6, 7)

(c) *F*(*x*, *y*, *z*) = ∑ (2, 3, 4, 5) (d) *F*(*x*, *y*, *z*) = ∑ (1, 2, 3, 5, 6, 7)

14. Simplify the following Boolean functions, using four-variable k-maps:

(a) *F* (*w*, *x*, *y*, *z*) = ∑ (1, 4, 5, 6, 12, 14, 15)

(b) *F* (*A*, *B*, *C*, *D*) = ∑ (*2, 3, 6, 7, 12, 13, 14*)

15. Simplify the following Boolean expressions, using four-variable maps:

(a) *w*’ *z* + *xz* + *x*’ *y* + *wx*’ *z*

(b) *AD*’ + *B*’*C*’*D* + *BCD*’ + *BC*’*D*

16. Simplify the following Boolean function *F* , together with the don’t-care conditions *d* , and

then express the simplified function in sum-of-minterms form:

(a) *F*(*x, y, z*) = ∑(0, 1, 4, 5, 6) (b) *F* (*A, B, C, D*) = ∑ (0, 6, 8, 13, 14)

*d(x, y, z)* = ∑(2, 3, 7) *d*(*A, B, C, D*) = ∑ (2, 4, 10)

17. Design the circuit then implement in Verilog for the following:

a. Half adder

b. Full adder using Half adder/ without using half adder

c. Ripple carry adder

d. Carry Look ahead adder

e. Adder subtractor together

f. BCD to 7 segment display with don’t care