Midterm Test

(Time allowed: 90 minutes)
March 23, 2025

NOTE: Answer ALL **8** questions. Show all intermediate steps, except Question 1 and Question 2.

1. (10 points) Are the following statements necessarily true (Y) or not (N)?

(1) A full-adder can be realized only by using 2-input AND gates;

(2) 11010101 is an 8-bit number in 2's complement form for a decimal number -42;

(3) For 2's complement signed numbers, the range of values for 5-bit numbers is from -15 to $+16$;

(4) The sum (BCD form) of two BCD numbers 1001 and 0101 is 1110;

(5) The odd parity bit for the BCD number 00010100 is 1;

(6) When '1' and '0' are input into a 2-input XNOR gate, the output is 1;

(7) Subtract 172_{16} from BCD_{16}, the result is $A5B_{16}$;

(8) The Gray code for the binary 101100 is 111011;

(9) A NOR gate can be substituted by an XOR gate;

(10) A ternary numeral system (also called base 3 or trinary) has three as its base. Analogous to a bit, a ternary digit is a trit (trinary digit). Ternary most often refers to a system in which the three digits are all non-negative numbers; specifically 0, 1, and 2. The decimal for the ternary number 101 is 5.

Solution:

(1) N; 2 NAND gates, 2 XOR gates and 1 OR gate;

(2) N; 11010110;

(3) N; -16 to $+15$;

(4) N; 00010100;

(5) Y; 0;

(6) N; 0;

(7) Y; $A5B_{16}$;

(8) N; 111010;

(9) N; negative-AND gate;

(10) N; 10.

2. (10 points)

(1) (3 points) Convert the binary number 1110001 into octal representation;

(2) (3 points) Perform the subtraction of the signed numbers: 00001100 − 11110110;

(3) (4 points) Divide the signed number 00011001 by the signed number 00000101.

Solution:

(1) $1110001_2 = 71_{16} = 161_8$;

(2) In this case, $13 - (-10) = 13 + 10 = 23$

$$\begin{array}{r} 00001100 \\ +00001010 \\ \hline 00010110 \end{array}$$

(3) In this case, $00000101 \rightarrow 11111011$ (2's complement)

$$\begin{array}{r} 00011001 \\ +11111011 \\ \hline 100010100 \\ +11111011 \\ \hline 100001111 \\ +11111011 \\ \hline 100001010 \\ +11111011 \\ \hline 100000101 \\ +11111011 \\ \hline 100000000 \end{array}$$

Quotient is $5 = 0101$. remainder is 0. Note that 1 indicates the result is positive. It is discarded.

3. (15 points)

(1) (7 points) In a biased N-bit binary number system with bias B, positive and negative numbers are represented as their value plus the bias B. For example, for 5-bit numbers with a bias of 15, the number 0 is represented as 01111, 1 as 10000, and so forth. Consider a biased 8-bit binary number system with a bias of 63_{10}. What decimal value does the binary number 10000010_2 represent?

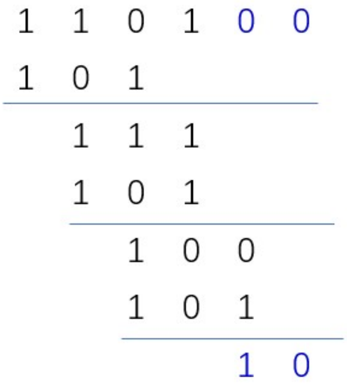
(2) (8 points) Apply CRC to the data bits 1101 using the generator code 101 to produce the transmitted CRC code. (Tips: The Number of zeros to append equals the length of the generator code minus one.)

Solution:

(1) $10000010_2 = 1 \times 2^7 + 1 \times 2^1 = 128 + 2 = 130$

130-63=67

(2) The remainder is 10, so the transmitted CRC is 110110



```
1 1 0 1 0 0
1 0 1
-----------
  1 1 1
  1 0 1
  -------
    1 0 0
    1 0 1
    -------
        1 0
```

Figure 1: Q3-2 solution.

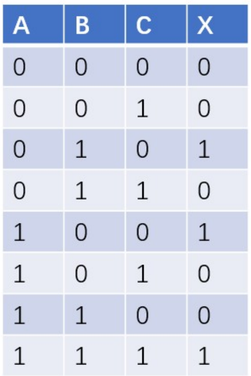
4. (15 points)

(1) (5 points) Develop a truth table for $ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C}$.

(2) (5 points) Convert $ABC + A\bar{B}C + \bar{A}B\bar{C}$ into the standard Product Of Sums (POS) form.

(3) (5 points) Convert $AB + CD + AD$ into the standard Sum Of Products (SOP) form.

Solution:



A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 2: Q4-1 Truth Table.

(1)

(2)

$$\begin{aligned}
&ABC + A\bar{B}C + \bar{A}B\bar{C}\\
=&A(BC + \bar{B}C) + \bar{A}B\bar{C}\\
=&AC + \bar{A}B\bar{C}\\
=&(AC + \bar{A})(AC + B\bar{C})\\
=&(\bar{A} + A)(\bar{A} + C)(AC + B)(AC + \bar{C})\\
=&(\bar{A} + C)(A + B)(B + C)(A + \bar{C})(C + \bar{C})\\
=&(\bar{A} + C + B\bar{B})(A + B + C\bar{C})(B + C + A\bar{A})(A + \bar{C} + B\bar{B})\\
=&(\bar{A} + C + B)(\bar{A} + C + \bar{B})(A + B + C)(A + B + \bar{C})(B + C + A)(B + C + \bar{A})(A + \bar{C} + B)(A + \bar{C} + \bar{B})\\
=&(\bar{A} + B + C)(\bar{A} + \bar{B} + C)(A + B + C)(A + B + \bar{C})(A + \bar{B} + \bar{C})
\end{aligned}$$

(3)

$$\begin{aligned}
&AB + CD + AD\\
=&AB(C + \bar{C}) + (A + \bar{A})CD) + AD(B + \bar{B}))\\
=&ABC + AB\bar{C} + ACD + \bar{A}CD + ABD + A\bar{B}D\\
=&ABC(D + \bar{D}) + AB\bar{C}(D + \bar{D}) + ACD(B + \bar{B}) + \bar{A}CD(B + \bar{B}) + ABD(C + \bar{C}) + A\bar{B}D(C + \bar{C})\\
=&ABCD + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D} + ABCD + A\bar{B}CD\\
&+ \bar{A}BCD + \bar{A}\bar{B}CD + ABCD + AB\bar{C}D + A\bar{B}CD + A\bar{B}\bar{C}D\\
=&ABCD + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D} + A\bar{B}CD + \bar{A}BCD + \bar{A}\bar{B}CD + A\bar{B}\bar{C}D
\end{aligned}$$

5. (12 points)

The gate labeled "K gate" has the truth table as shown in Table 1.

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Table 1: The truth table of K gate

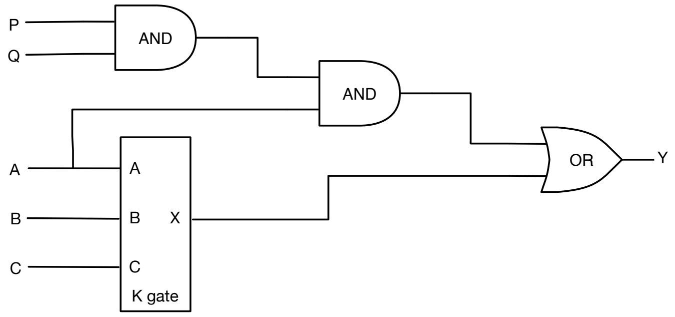


Figure 3: Q5 circuit.

(1) (5 points) Write down the logic expression in the **standard sum of product form**;

(2) (7 points) Given the circuit as shown in Figure 3, analyze this circuit and come-up with the logic circuit for the same function that uses **ONLY** NAND gates (The complementary inputs are not directly available - e.g., $\bar{A}$ is not directly available.)

Solution:

(1) $X = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$

(2)

$$\begin{aligned} X =& \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC \\ =& (\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}) + (AB\bar{C} + ABC) \\ =& (\bar{A}\bar{C} + A\bar{C}) + AB \\ =& \bar{C} + AB \end{aligned}$$

$Y = PQA + X = APQ + AB + \bar{C}$

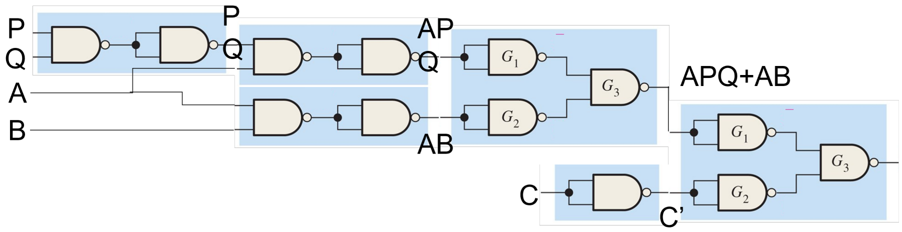


Figure 4: Q5-2 solution.

6. (15 points) Design a circuit with four inputs, namely A, B, C, and D, and one output Y. The output is 1 when the consecutive sequence '110' is present, e.g., 0110.

(1) (3 points) Develop the truth table for Y with four inputs A, B, C and D;

(2) (5 points) Use a Karnaugh map to find the minimum product of sums (POS) form of Y;

(3) (7 points) Implement the logic function specified in the truth table of (1) by using a 4-bit Decoder and basic logic gates.

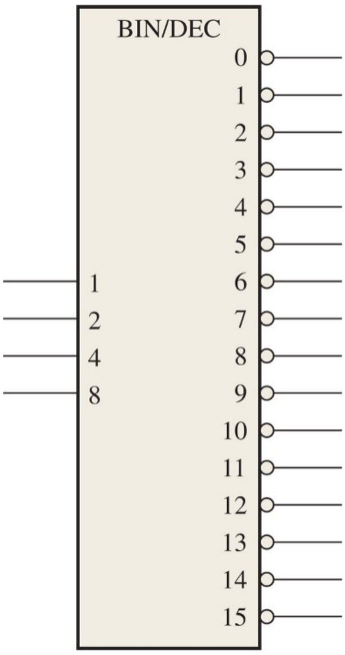


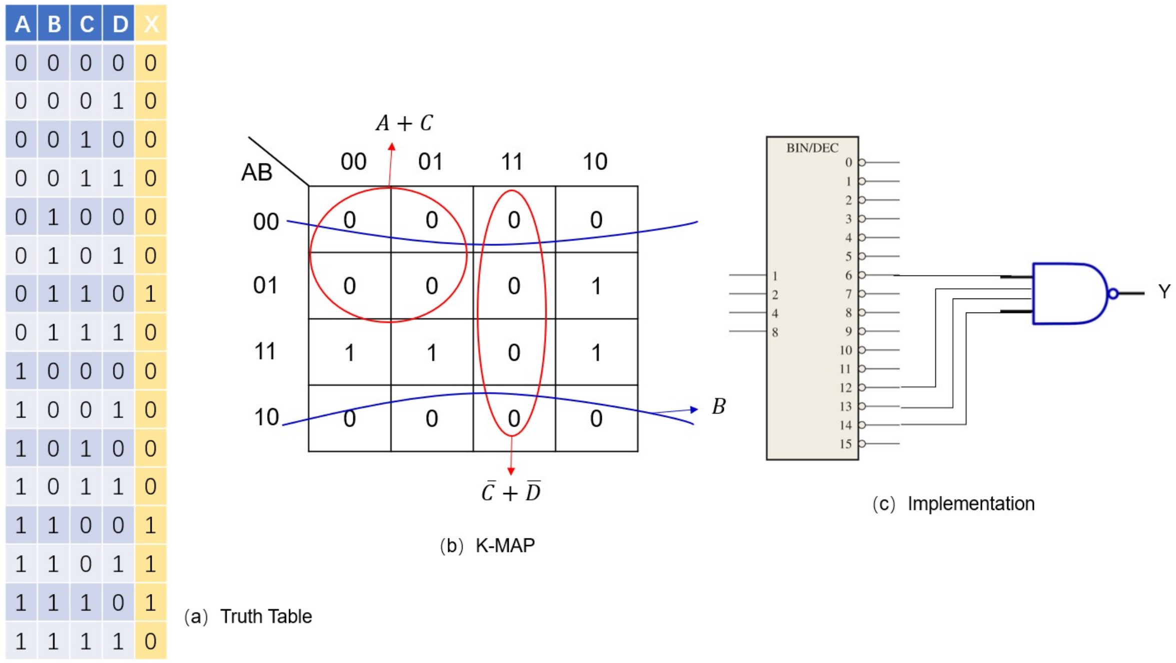
Figure 5: 4-bit Decoder.

Solution:

(1) Fig.6(a).

(2) Fig. 6(b).
$Y = B(\bar{C} + \bar{D})(A + C)$

(3) Fig. 6(c).



A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

(a) Truth Table

AB	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	1	0	1
10	0	0	0	0

(b) K-MAP

(c) Implementation

Figure 6: Q6 solution.

7. (13 points) A ripple carry adder is one in which the carry output of each full adder is connected to the carry input of the next higher-order stage (a stage is one full adder). The sum and the output carry of any stage cannot be produced until the input carry occurs; this causes a time delay in the addition process, as shown in Figure 7. The carry propagation delay for each full adder is the time from the application of the input carry until the output carry occurs, assuming that the A and B inputs are already present.

Full-adder 1 (FA1) cannot produce a potential output carry until an input carry is applied. Full-adder 2 (FA2) cannot produce a potential output carry until FA1 produces an output carry. Full-adder 3 (FA3) cannot produce a potential output carry until an output carry is produced by FA1 followed by an output carry from FA2, and so on. As shown in Figure 7, the input carry to the least significant stage has to ripple through all the adders before a final sum is produced. The cumulative delay through all the adder stages is a "worst-case" addition time. The total delay can vary, depending on the carry bit produced by each full adder. If two numbers are added such that no carries (0) occur between stages, the addition time is simply the propagation time through a single full adder from the application of the data bits on the inputs to the occurrence of a sum output; however, worst-case addition time must always be assumed.

Consider multiplication with unsigned 4-bit numbers can be performed as shown in Figure 11.

(1) (10 points) Implement this combinational multiplier by using full adders and AND gates **ONLY**.

(2) (3 points) The critical path is the longest path in the circuit. Propagation delay (latency) is one of the important factors to describe a digital circuit. It is defined as the time needed for an input change to produce an output change. The critical path is the path in the entire design with the maximum propagation delay. As an example, a critical path is denoted in Figure 7. Analyze whether there is a possible critical path in your designed combinational multiplier. If yes, please mark the critical path and approximate its propagation delay (Assume the propagation delay of each adder is t, and the propagation delay of AND gate is $0.1t$); If no, please give reasons.

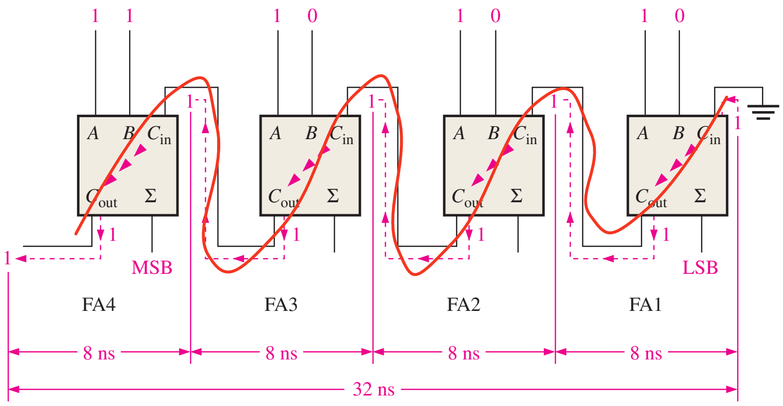


Figure 7: A 4-bit parallel ripple carry adder with a critical path.

```
                    X3    X2    X1    X0   ← multiplicand
                 *  Y3    Y2    Y1    Y0   ← multiplier
                 --------------------
                     X3Y0  X2Y0  X1Y0  X0Y0  ⎫
+               X3Y1  X2Y1  X1Y1  X0Y1       ⎬ Partial products, one for
+         X3Y2  X2Y2  X1Y2  X0Y2             ⎪ each bit in multiplier
+   X3Y3  X2Y3  X1Y3  X0Y3                   ⎭
-----------------------------------------
     Z7    Z6    Z5    Z4    Z3    Z2    Z1    Z0
```

Figure 8: Multiplication with unsigned 4-bit numbers.

Solution:

(1) The combinational multiplier is shown in Figure 9.

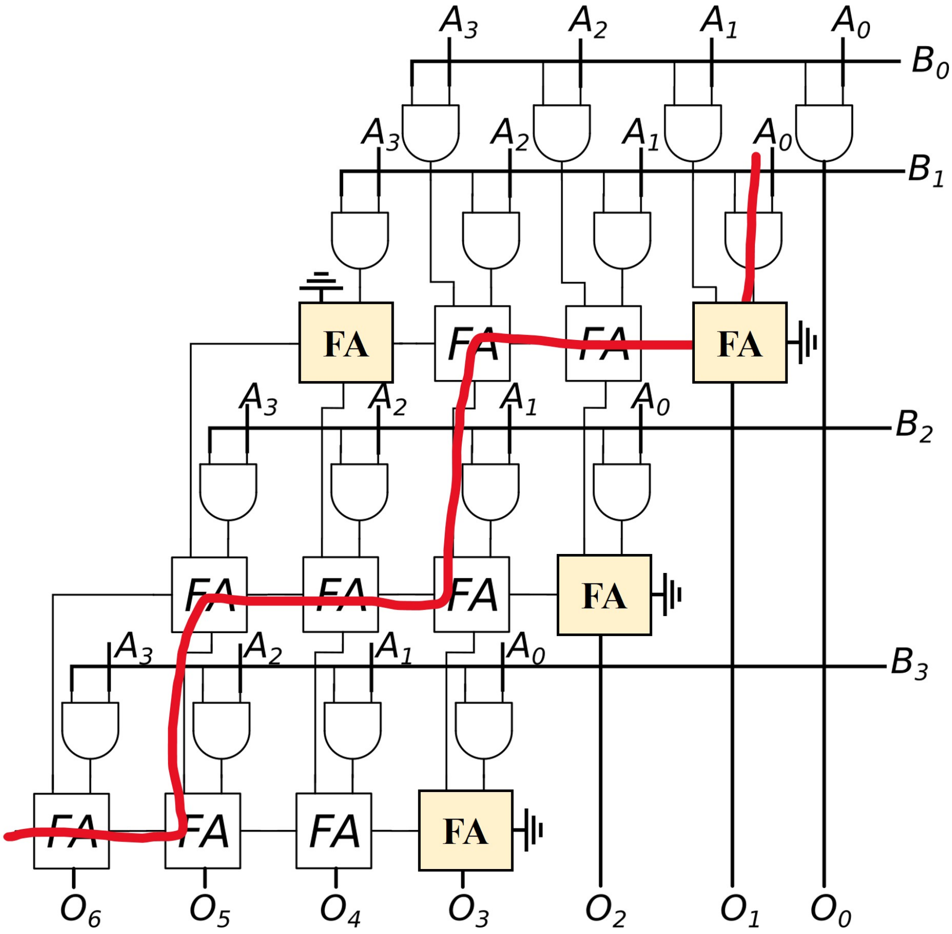


Figure 9: Q7 multiplier.

(2) The propagation delay is $8.1t$.

8. (10 points) The IEEE 754 standard specifies a 32-bit binary as having: 1 bit Sign; 8 bits Exponent width; 24 bits (23 explicitly stored) Mantissa precision. The sign bit determines the sign of the number, which is the sign of the mantissa as well. The exponent is an 8-bit unsigned integer from 0 to 255, in biased form: an exponent value of 127 represents the actual zero. Exponents range from -126 to +127 because exponents of -127 (all 0s) and +128 (all 1s) are reserved for special numbers. The true mantissa includes 23 fraction bits to the right of the binary point and an implicit leading bit (to the left of the binary point) with value 1, unless the exponent is stored with all zeros. Thus only 23 fraction bits of the mantissa appear in the memory format, but the total precision is 24 bits. The bits are laid out as shown in Figure 10: The real value assumed by a given 32-bit binary data with a given sign, biased exponent (the 8-bit unsigned integer), and a 23-bit mantissa is

$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\cdots b_{23})_2-127} \times (1.b_{22}b_{21}\cdots b_0)_2.$$

(1) (4 points) Convert the decimal number 8.75_{10} to IEEE 754 standard 32-bit binary. Show your work;

(2) (4 points) A IEEE 754 customization specifies a 16-bit binary as having: 1 bit Sign; 4 bits Exponent width; 12 bits (11 explicitly stored) Mantissa precision. The sign bit determines the sign of the number. The exponent is a 4-bit unsigned integer from 0 to 15, in the biased form: an exponent value of 7 represents the actual zero. Exponents range from -6 to +7 because exponents of -7 (all 0s) and +8 (all 1s) are reserved for special numbers. The true mantissa includes 11 fraction bits to the right of the binary point and an implicit leading bit (to the left of the binary point) with value 1, unless the exponent is stored with all zeros. Thus only 11 fraction bits of the mantissa appear in the memory format, but the total precision is 12 bits. Sketch the IEEE 754 customization 16-bit binary layout;

(3) (2 points) Convert the decimal number 8.75_{10} to the IEEE 754 customization 16-bit binary. Show your work.

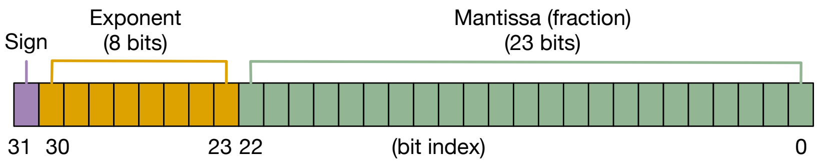


Figure 10: IEEE 754 standard 32-bit binary.

Solution:

(1) $8.75_{10} = 1000.11_2 = 1.00011 \times 2^{011}$; $011 + 1111111 = 10000010$; 0 10000010 00011000000000000000000;

(2) IEEE 754 standard 16-bit binary layout is shown below;

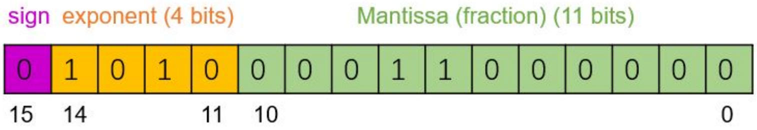


Figure 11: IEEE 754 standard 16-bit binary.

(3) $8.75_{10} = 1000.11_2 = 1.00011 \times 2^{011}$; $011 + 111 = 1010$;
0 1010 00011000000.