(20 points) (5 points) Problem 1 –

A. (5 points) Convert 23.625\(_{10}\) to binary

B. (5 points) Convert 11011001.110101\(_{2}\) to decimal

C. (2.5 points) Convert DEADBEEF\(_{16}\) to binary

D. (7.5 points) The following numbers are represented in decimal and the corresponding 2’s complement representations. Perform the following additions and subtractions. Explain the results (correct, incorrect, overflow, etc) using your own words.

\[
\begin{array}{cccccc}
+6 & 0110 & -8 & 1000 & +3 & 0011 \\
+3 & 1101 & +8 & 1000 & -4 & 1100 \\
\end{array}
\]
Problem 2 (20 points) (5 points) –

A. (10 points) Given the function \( F(A, B, C) = A'\cdot B'\cdot C + A'\cdot B\cdot C' + A\cdot B'\cdot C' + A\cdot B\cdot C \), fill in the truth table and draw a logic gate implementation, using a minimal number of gates.

Draw the output waveform for the circuit below. The reference grid in the timing diagram is one line per 10 units. Considering 10 units=2 seconds what is the clock period of signal A (\( T_A \)), signal B (\( T_B \)) and Signal C (\( T_C \)). What is the frequency of signal A?

B. (5 points) Convert the function \( Z(A, B, C) = A'\cdot (B+C) + B'(A+C') \) to minterm canonical form using the \( \Sigma \) notation.

C. (5 points) Write the complement of the function \( Z(A, B, C, D) = \Sigma(0,1,4,8,13,15) \) in maxterm (\( \Pi \)) form.
Problem 3 (20 points) (5 points) –

A. (5 points) Convert $Z(A,B,C,D) = \Sigma(0, 2, 3, 4, 6, 9, 10, 12, 14)$ to minimal And–Or logic (Sum-of-Product). No implementation required.

B. (7.5 points) Give the hazard free form of $Z(A,B,C) = \Sigma(1, 3, 4, 6, 7)$. Implement the hazard free function, in NAND-NAND form.

A. (7.5 points) Convert $Z(A,B,C,D) = \Sigma(2, 3, 6, 7, 12, 13, 15)$ to minimal Product-of-Sum form. Implement the function in NOR-NOR form.
Problem 4 (20 points) (5 points)

A. (5 points) Implement the logic function \( Z(A, B, C) = A' \cdot (B+C) + B'(A+C') \) using the 3 to 8 decoder shown below. C is the most significant selection bit of the decoder. Use additional gates if necessary.

B. (5 points) Implement the function \( r(e,f,g,h) = \Sigma(0,3,5,6,9,10,12,15) \) using the 8 to 1 multiplexer shown below. C is the most significant selection bit of the multiplexer. The output of the multiplexer is \( Y \). Use logic gates if necessary.

C. (5 points) Make the necessary connections to the 74LS163 counter to operate in the free running mode (from 0 to 15 and back). The 74LS163 is a synchronous 4 bit binary counter. It has \( P \) and \( T \) inputs which enable counting when high, a clock input, a parallel load input \( (D-A) \) which is used when the \( LOAD \) input is low and a the clock. The clear input loads zero into the counter and is also synchronous and active low. The \( RCO \) output goes high when the output is 15. The outputs, \( QD-QA \), reflect the state of the counter.

D. (5 points) Use the 74L194 to build a basic Johnson counter. An n-bit shift register with the complement of the serial output fed back into the serial input is a counter with \( 2^n \) states and is called twisted ring, Moebius or Johnson counter. The universal register 74LS94 uses a clock input, an active low clear input and two mode inputs. The modes are \( S1S0=00 \), hold, \( S1S0=01 \) shift right (toward \( D \)), \( S1S0=10 \) shift left (toward \( A \)) and \( S1S0=11 \), parallel load. \( R \) is the serial input for right shift and \( L \) is the serial input for left shift.
Problem 5 (20 points) (5 points)

A. (7.5 points) Analyze the following circuit. Write the excitation and output equations, fill in the transition table and draw a state diagram for the state machine shown below. You may ignore the set and reset inputs for the purpose of your state diagram.

B (7.5 points) Design using JK flip-flops a synchronous binary counter that counts from 0 to 7 and back.

C (5 points) Draw a state diagram for a state machine that sets output to logic 1 whenever two 1s in a row are found in the input sequence, and the output stays in logic 1 until two 0s are found in the input sequence, at which time the output changes to zero again. Name the states a, b, c, d and assume past sequence of all 0's upon power up. Do not implement the circuit.