
Digital Design with Verilog

Date: June 3, 2025

Review of Combinational and Sequential Circuits

1. Consider an 8-bit ripple-carry adder built using eight full-adders. Each full adder is built using a two-level sum-of-products circuit for the carry-out and a 3-input XOR gate for the sum. In this adder assume that the delay through a gate is:

Gate delay = $1 + 0.1 \times (\# \text{Inputs} - 1)$ ns,
where # Inputs is the number of inputs to the gate.

Using this equation, the delay through an inverter is equal to 1 ns, the delay through a 2-input gate (any type) is 1.1 ns, and so on.

- (a) What is the critical path delay in the ripple-carry adder? (Recall: the critical path is the longest (slowest) path in the circuit.) Give your answer in ns.
 - (b) Consider next a carry lookahead adder, as discussed in class. If you can use gates of any size (number of inputs), what is the critical path delay in the 8-bit carry lookahead adder?
 - (c) Now assume that you can use gates with a fanin of no more than four inputs. What is the critical path delay for the carry lookahead adder in this case?
2. Design a four-bit combinational 2's complementer circuit. (The output generates the 2's complement of the input binary number.) Show that the circuit can be constructed with exclusive-OR gates. Can you predict what the output functions are for a five-bit 2's complementer?
 3. This question is about building a digital circuit that will add three 4-bit numbers, X , Y , and Z . The individual bits of these three numbers will be represented as X_i , Y_i , and Z_i . To illustrate this clearly, the addition of three 4-bit numbers would be written on papers as follows:

		x_3	x_2	x_1	x_0
	+	y_3	y_2	y_1	y_0
	+	z_3	z_2	z_1	z_0
s_5	s_4	s_3	s_2	s_1	s_0

block to create an adder that adds two N-bit numbers. You will design the complete circuit in two steps, part (a) and (b) below.

- (a) Using only FA building blocks give the design of a new building block that performs the function needed to implement the box surrounding x_2 , y_2 , z_2 , and s_2 , (shown above). You will call this building block TNFA and use it in part (b) below. It computes the sum bit S_i , corresponding to the three inputs X_i , Y_i , and Z_i and any other inputs and outputs that are necessary in the context of the full adder.
 - (b) Using the building block of part (a), the TNFA, and any other logic gates you deem necessary give the design of the three 4-bit number adder (which produces a 6-bit sum, $s_5 \dots s_0$). Your answer should show all the inputs (i.e. all X_i , Y_i and Z_i , $i = 0 \dots 3$ and any carry inputs) and outputs necessary to make the complete adder function correctly.
4. Design a combinational circuit whose input A is a vector of 7-bits and its output Z is a 3-bit vector which counts the number of "1"s present in the input. For instance, $A = 1001001$ or $A = 1010100$ both will results in $Z = 011$, indicating that both input vectors have three numbers of "1"s. *Please note that you have to use only full-adders while building this circuit.*

5. In computer arithmetic, it is often necessary to perform comparisons. The logic diagram shown in Figure 1 is a four-bit adder/subtractor circuit that can be used for comparison by performing $A - B$, where $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$. The three outputs of the circuit denote the following:

- $Z = 1$ if the result is 0; otherwise $Z = 0$
- $N = 1$ if the result is negative; otherwise $N = 0$
- $V = 1$ if arithmetic overflow occurs; otherwise $V = 0$

Show how Z , N , and V can be used to determine the cases $A = B$, $A < B$, $A \leq B$, $A > B$, and $A \geq B$.

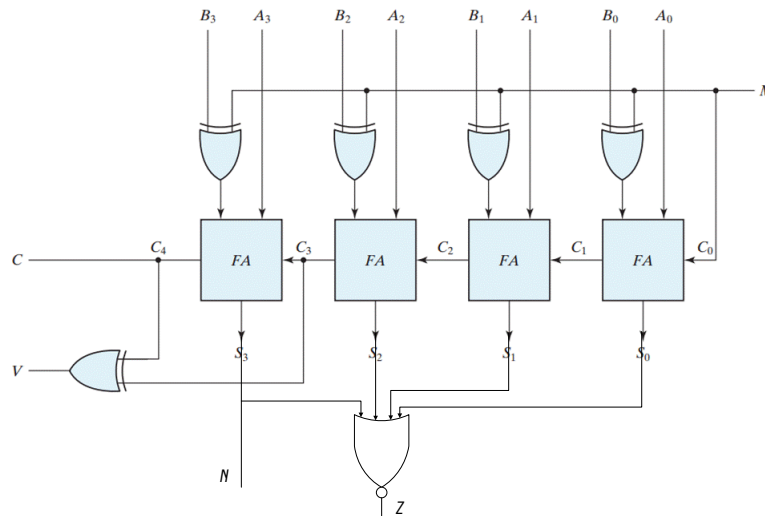


Figure 1: Four-bit adder/subtractor for performing the comparison.

6. The following circuit shown in Figure 2 will compare two binary numbers A and B each has two bits (A_1A_0) and (B_1B_0) then will output 1 if the two numbers are equal, and zero otherwise. Design the circuit using an 8-to-1 multiplexer.

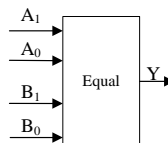


Figure 2: Two-bit equality comparator

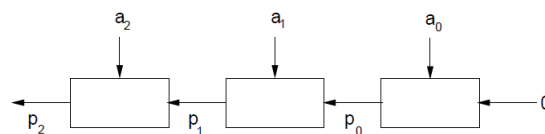
7. Design a circuit that takes in a BCD digit and produces a two digit BCD number. This two digit number is nothing but input BCD digit multiplied by 5.

8. Complete the following table.

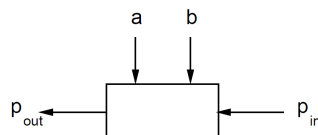
Representation			
Decimal	Signed Magnitude	1's complement	2's complement
	1110		
		1111	
			100
		01110	

Table 1: Number conversions

9. Sketch the circuit that computes $|A - B|$, where A, B are 4-bit unsigned numbers. For example, $A = 0101$, $B = 1101 \rightarrow |A - B| = |5 - 13| = 8$. You can only use full adders (or multi-bit adders) and logic gates. Your circuit must avoid overflow: design your circuit so that the result and intermediate operations have the proper number of bits.
10. We want to build a NAND gate circuit to compute the parity of an n -bit unsigned number. The parity is defined as 1 if and only if there are an odd number of 1's in the number. One way of doing this is to build the circuit 1 bit at a time (as in the adder), such that the circuit computes the parity after that bit as a function of the parity up to that bit and the one input bit.
- (a) A block diagram of the first few bits of such circuit is shown below. Show an NAND gate circuit to implement 1-bit and compute the delay of n -bits. Assume that the delay for an NAND is $1 + 0.1 * (\# \text{ of inputs}) \text{ ns}$



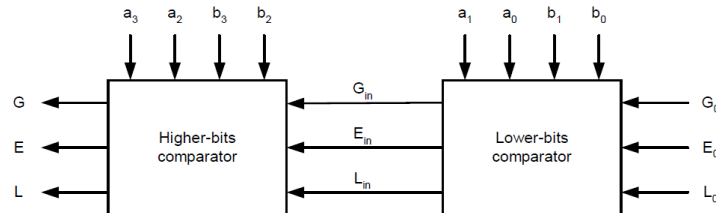
- (b) Reduce the delay by implementing 2-bits at a time as shown below. Draw the NAND gate circuit. Compute the delay for n -bits in this case.



11. Using a 2×1 multiplexer, build NOT, AND, OR, D-latch, and D-FlipFlop.
12. Design/sketch your idea for a 4-bit magnitude comparator that works for both signed and unsigned numbers. Your circuit should use minimum number of logic gates.
13. A digital comparator is a logic circuit that takes two n -bit numbers A and B as input and compares the magnitude of A and B . It has three outputs, G , E , and L , where $G=1$ only if $A > B$, $E = 1$ only if

$A = B$, and $L = 1$ only if $A < B$.

A 4-bit magnitude comparator (for unsigned numbers) is to be constructed from two 2-bit comparators as shown in the diagram below. One of these will compare the high-order 2-bits (a_3a_2 and b_3b_2) of each input and the other will compare the lower 2-bits (a_1a_0 and b_1b_0). The outputs, greater (G_{in}), equal (E_{in}), and less (L_{in}) from the lower-bits comparator become additional inputs to the higher-bits comparator. The inputs to the lower-bits comparator are labeled G_0 , E_0 , and L_0 .



- (a) For what values of the inputs A and B will the result of the lower-bits comparator (G_{in} , E_{in} and L_{in}) influence the overall outputs G, E and L? Explain. Your answer **should not** give every possible input combination—you must express your answer by stating relationships between the input values of the a_i , b_i , G_{in} , E_{in} , and L_{in} .
 - (b) If both 2-bit comparator units are to contain identical logic circuits, specify, for the lower-bits comparator, what values the inputs G_0 , E_0 , and L_0 should have in order for the complete 4-bit comparator to operate properly.
 - (c) Give a logic expression for G (recall that $G = 1$ if $A > B$) in terms of a_3 , a_2 , b_3 , b_2 and G_{in} . You are allowed to use the AND, OR, NOT and XOR logic operators as needed.
 - (d) Give a logic expression for L (recall that $L = 1$ if $A < B$) in terms of a_3 , a_2 , b_3 , b_2 and L_{in} . You are allowed to use the AND, OR, NOT and XOR logic operators as needed.
 - (e) Give a logic expression for E (recall that $G = 1$ if $A = B$) in terms of a_3 , a_2 , b_3 , b_2 and E_{in} . You are allowed to use the AND, OR, NOT and XOR logic operators as needed.
14. Using only 2:1 mux structures, design a block that compares two 8-bit values (unsigned, simple 0-255 values) – A,B. if $A > B$ then the output is 1 else 0. Try to use minimum amount of muxes.
 15. For the state table shown in Figure 1.
 - (a) Draw the corresponding state diagram.
 - (b) Tabulate the reduced state table.
 - (c) Draw the state diagram corresponding to the reduced state table.
 16. Starting from state a, and the input sequence 01110010011, determine the output sequence for:
 - (a) The state table of the previous problem.
 - (b) The reduced state table from the previous problem. Show that the same output sequence is obtained for both.
 17. Design a sequential circuit with two D flip-flops A and B, and one input x_{in} .
 - (a) When $x_{in} = 0$, the state of the circuit remains the same. When $x_{in} = 1$, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.
 - (b) When $x_{in} = 0$, the state of the circuit remains the same. When $x_{in} = 1$, the circuit goes through the state transitions from 00 to 11, to 01, to 10, back to 00, and repeats.
 18. Design a one-input, one-output serial 2's complemener. The circuit accepts a string of bits from the input and generates the 2's complement at the output. The circuit can be reset asynchronously to start and end the operation.

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

Figure 3: State Table

19. Design the sequential circuit for the state diagram in Figure 2, using T flip-flops.

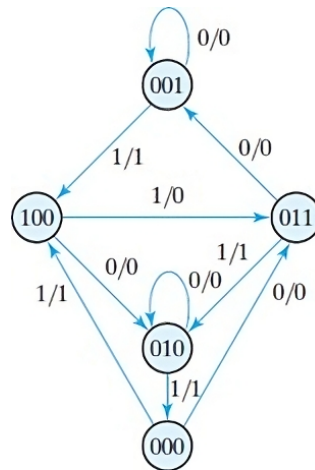


Figure 4: State Diagram

20. Realize serial adder using Moore and Mealy machines. Serial adder is a digital circuit in which bits are added one pair at a time. The block diagram of the serial adder is shown in Figure 3.

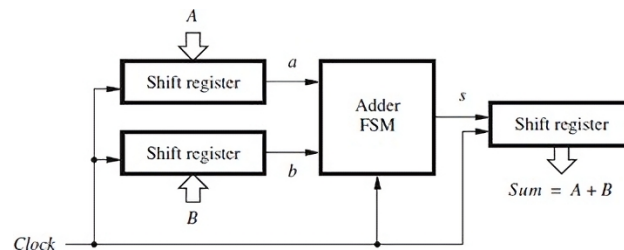


Figure 5: Serial adder

21. For the ripple counter in Figure 6, show the complete timing diagram for sixteen clock pulses. show the clock, Q_0 , Q_1 , and Q_2 waveforms.

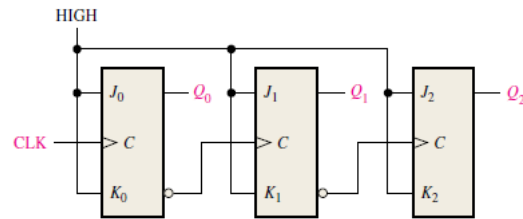


Figure 6: Three-bit ripple counter

22. Design an FSM for a vending machine as per the description given below.

Description: A vending machine is an automated machine that provides items such as snacks, chocolates, ice creams and cold drinks to consumers after money, a credit card, or a specially designed card is inserted into the machine. The vending machine in the present case of consideration sells bottles of water for 75 cents. Design a Vending Machine which accepts dollars (worth 100 cents) and quarters (worth 25 cents). Once a sufficient amount of money is entered the vending machine will dispense a bottle of water. If the user entered a dollar it will return one quarter in change. A “Money Receiver” detects when money has been entered. The receiver sends two signals to FSM indicating whether a dollar or quarter has been received. A “Bottle Dispenser” system holds the water bottles and release one water bottle when its input signal is asserted. A “Coin Return” system holds quarters for change and release one quarter when its input signal is asserted. The money receiver will reject money if a dollar and quarter are entered simultaneously or if a dollar is entered once the user has started entering quarters.

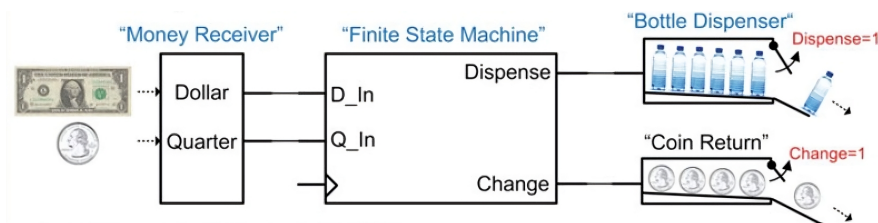
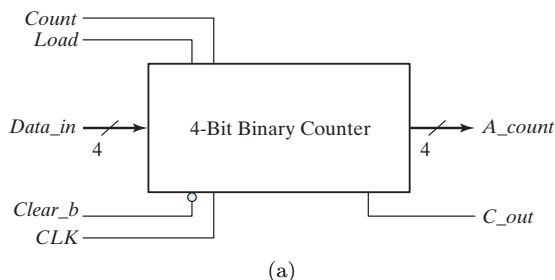


Figure 7: Vending machine

23. Implement D-Latch using multiplexers and transmission gates.
24. Implement both positive edge-triggered and negative edge-triggered D-Flip-Flops using multiplexers and transmission gates.
25. Design a four-bit binary counter with parallel load as per the block diagram and functionality given in Figure 8. Please note that C_out is an output for the next stage.



Clear_b	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

(a)

(b)

Figure 8: 4-bit parallel counter with load