ECE 2300 Digital Logic and Computer Organization

Topic 8: Sequential Building Blocks

http://www.csl.cornell.edu/courses/ece2300
School of Electrical and Computer Engineering
Cornell University

revision: 2025-12-09-12-40

List of Problems

Problem 1. Comparative Analysis of Multiplier Designs

In this problem, we will consider four different implementations of an unsigned two-input integer multiplier capable of multiplying a 32-bit operand (input A) by a 4-bit operand (input B) to produce a truncated 32-bit result (output Z).

We compute the multiplication by performing multiple additions, exploiting the fact that an operand can be decomposed into a sum. This sum can then be multiplied term-by-term with the other operand, and the resulting products can be added together. We decompose operand B into a sum of powers of two (corresponding to B's bits), since multiplying by each power of two (1, 2, 4, 8, ...) corresponds to a simple bit-shift operation. For example, multiplying A by two is equivalent to left-shifting A by one bit position; multiplying by four corresponds to a left-shift by two positions, and so on.

For instance, if $B = 7$ (binary: 0111; decimal: $4 + 2 + 1$), the result is computed as the sum of $A \times 4$ (A left-shifted by two bits), $A \times 2$ (A left-shifted by one bit), and $A \times 1$ (A unshifted).

We will fill the following table during this assignment. For each design, we compute the average number of clock cycles per transaction, the clock period, the total execution time for 100 transactions, and lastly the area of the design.

Microarchitecture	**Num Trans (#)**	**Avg Cycles per transaction (cyc/trans)**	**Clock Period (τ)**	**Total Execution Time (τ)**	**Area (α)**
Single-Cycle (Linear)	100				
Single-Cycle (Tree)	100				
Pipelined	100				
Multi-Cycle	100				

Use the parameters to the right for all following problems.

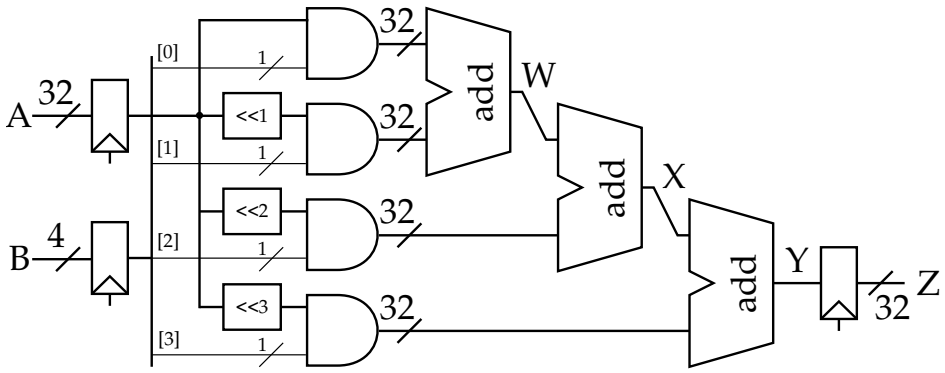
	t_{pd}	Area
Shift	0τ	0α
1-Bit AND2	3τ	3α
1-Bit 2-to-1 Mux	8τ	11α
32-Bit Ripple-Carry Adder	348τ	992α
1-Bit FF (t_{cq})	9τ	33α

FF (t_{setup})	10τ
FF (t_{hold})	1τ

Part 1.A Single-Cycle Linear Multiplier

We begin with the following linear single-cycle multiplier design, which contains three concatenated adders. The inputs are *A* (with a width of 32 bits) and *B* (with a width of 4 bits). Since shift and slice operations can be hardwired, we do not add any delay or area penalty for these modules. The AND gates directly preceding the adder modules are not typical AND gates. Upon closer inspection, one can observe that one input is 32 bits wide (input *A*), while the other (input *B*) is only a single bit. Thus, each *AND* gate actually consists of many AND2 gates in parallel, with each gate receiving the single *B* input bit and one of the input bits from *A*.

Note: When computing the area, consider whether all 32 AND2 gates are actually needed for each AND gate when its respective input A is shifted.



Fill out the simulation table for the single-cycle linear multiplier.

Simulation Table

Cycle	Input	After Reg	W	X	Y	Z
0	TA: A=12, B=3					
1	TB: A=7, B=8					
2	TC: A=24, B=0					
3						
4						

Fill out transaction diagram for the single-cycle linear multiplier.

Transaction Diagram

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Transaction A																					
Transaction B																					
Transaction C																					

Identify the critical path of the single-cycle linear multiplier. Draw its path in the block diagram. What is the minimum clock period (T_C) that would still ensure correct operation?

How many cycles are needed in average by the single-cycle linear multiplier to process a single transaction?

Compute the total execution time to process 100 transactions with the single-cycle linear multiplier (in units of τ). *Note: Use the following formula. Also consider the number of clock cycles until the first result appears at the output.*

$$\frac{\text{Time}}{\text{Sequence}} = \frac{\text{Transactions}}{\text{Sequence}} \times \frac{\text{Avg Cycles}}{\text{Transaction}} \times \frac{\text{Time}}{\text{Cycle}}$$

Compute the area of the single-cycle linear multiplier (in units of α).

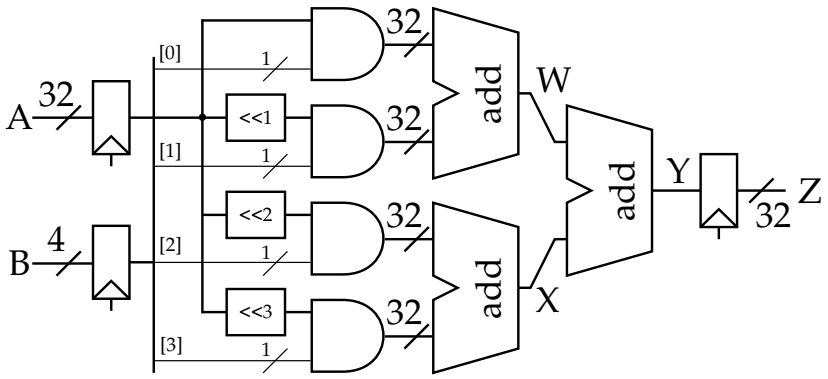
Add your computed data from the single-cycle linear multiplier to the comparison table above.

Nicholas's Solution

`https://vod.video.cornell.edu/id/1_k92jzikh`

Part 1.B Single-Cycle Tree Multiplier

We converted the single-cycle multiplier from the previous section into a tree structure. Utilize the same timing parameters as before.



Fill out the simulation table for the single-cycle tree multiplier.

Simulation Table

Cycle	Input	After Reg	W	X	Y	Z
0	TA: A=12, B=3					
1	TB: A=7, B=8					
2	TC: A=24, B=0					
3						
4						

Fill out transaction diagram for the single-cycle tree multiplier.

Transaction Diagram

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Transaction A																					
Transaction B																					
Transaction C																					

Identitfy the critical path of the single-cycle tree multiplier. Draw its path in the block diagram. What is the minimum clock period (T_C) that would still ensure correct operation?

How many cycles are needed in average by the single-cycle tree multiplier to process a single transaction?

Compute the total execution time to process 100 transactions with the single-cycle tree multiplier (in units of τ). *Note: Use the following formula. Also consider the number of clock cycles until the first result appears at the output.*

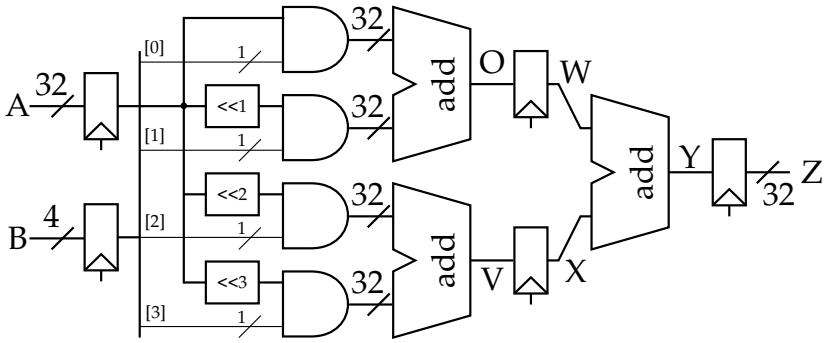
$$\frac{\text{Time}}{\text{Sequence}} = \frac{\text{Transactions}}{\text{Sequence}} \times \frac{\text{Avg Cycles}}{\text{Transaction}} \times \frac{\text{Time}}{\text{Cycle}}$$

Compute the area of the single-cycle tree multiplier (in units of α).

Add your computed data from the single-cycle tree multiplier to the comparison table above.

Part 1.C Pipelined Multiplier

Next, we introduce pipeline registers in our previous multiplier design.



Fill out the simulation table for the pipelined multiplier.

Simulation Table

Cycle	Input	After Reg	O	V	W	X	Y	Z
0	TA: A=12, B=3							
1	TB: A=7, B=8							
2	TC: A=24, B=0							
3								
4								

Fill out transaction diagram for the pipelined multiplier.

Transaction Diagram

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Transaction A																					
Transaction B																					
Transaction C																					

Identitfy the critical path of the pipelined multiplier. Draw its path in the block diagram. What is the minimum clock period (T_C) that would still ensure correct operation?

How many cycles are needed in average by the pipelined multiplier to process a single transaction?

Compute the total execution time to process 100 transactions with the pipelined multiplier (in units of τ). *Note: Use the following formula. Also consider the number of clock cycles until the first result appears at the output.*

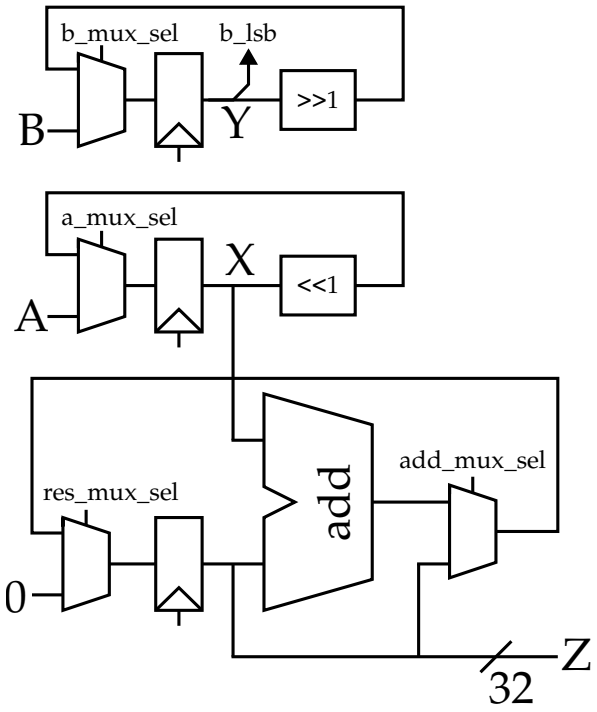
$$\frac{\text{Time}}{\text{Sequence}} = \frac{\text{Transactions}}{\text{Sequence}} \times \frac{\text{Avg Cycles}}{\text{Transaction}} \times \frac{\text{Time}}{\text{Cycle}}$$

Compute the area of the pipelined multiplier (in units of α).

Add your computed data from the pipelined multiplier to the comparison table above.

Part 1.D Multi-Cycle Multiplier

Lastly, we build a multi-cycle multiplier. Its datapath is shown below. The datapath contains a single adder. The control unit (not shown) manages the data to perform the multiplication over multiple cycles. Initially, it loads A and B through multiplexers into X and Y, respectively. Similarly, it initializes the result Z to zero. Each cycle, X is shifted one bit to the left (effectively doubling X) while Y is shifted to the right. If the current lowest bit of Y is set (b_lsb signal), the control unit adds X to Z (by setting the select signals of the muxes accordingly).



Fill out the simulation table for the multi-cycle multiplier.

Simulation Table

Cycle	Input	X	Y	Z
0	TA: A=12, B=3			
1				
2				
3				
4				
5	TB: A=7, B=8			
6				
7				
8				
9				
10				

Fill out transaction diagram for the multi-cycle multiplier.

Transaction Diagram

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Transaction A																					
Transaction B																					
Transaction C																					

Identitfy the critical path of the multi-cycle multiplier. Draw its path in the block diagram. What is the minimum clock period (T_C) that would still ensure correct operation?

How many cycles are needed in average by the multi-cycle multiplier to process a single transaction?

Compute the total execution time to process 100 transactions with the multi-cycle multiplier (in units of τ). *Note: Use the following formula. Also consider the number of clock cycles until the first result appears at the output.*

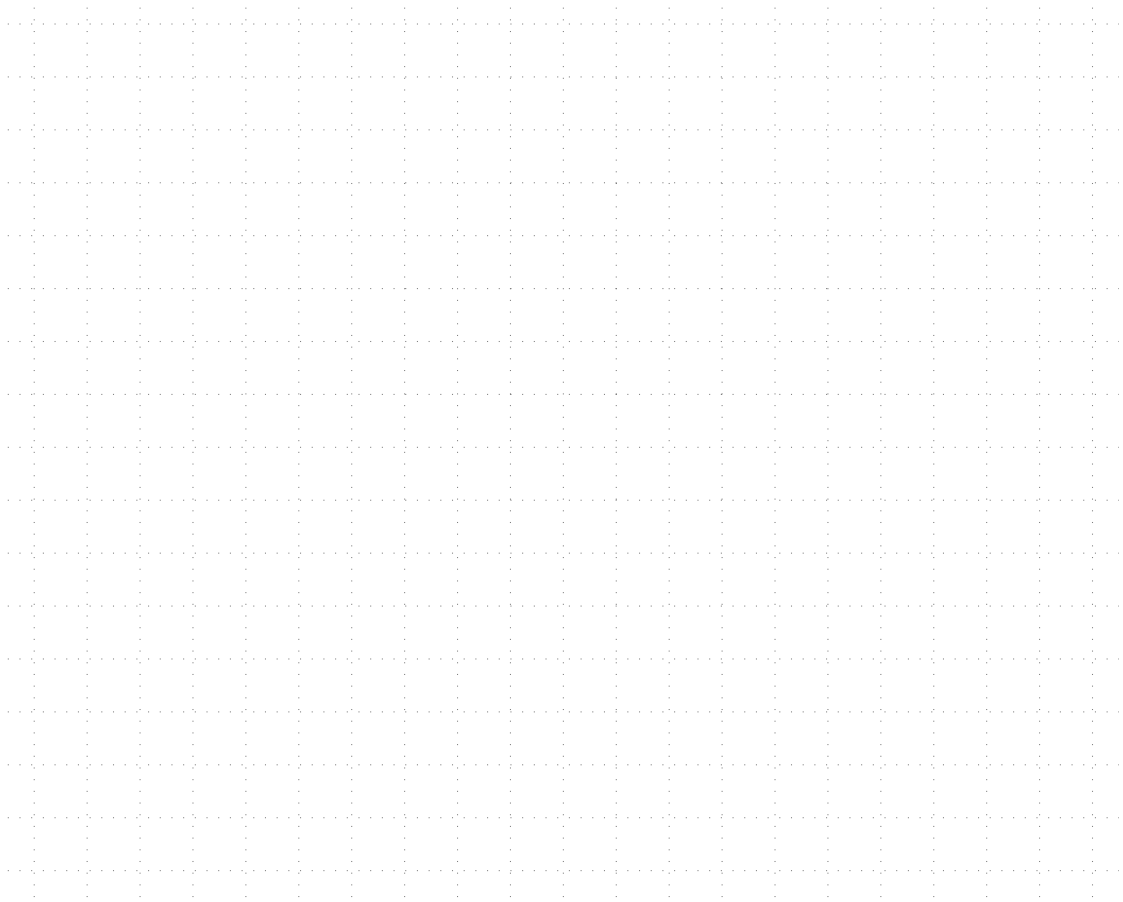
$$\frac{\text{Time}}{\text{Sequence}} = \frac{\text{Transactions}}{\text{Sequence}} \times \frac{\text{Avg Cycles}}{\text{Transaction}} \times \frac{\text{Time}}{\text{Cycle}}$$

Compute the area of the multi-cycle multiplier (in units of α).

Add your computed data from the multi-cycle multiplier to the comparison table above.

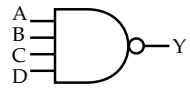
Part 1.E Comparative Analysis

Create a Pareto frontier plot of the four multiplier designs. *Note: X-Axis is area and Y-Axis is execution time.*



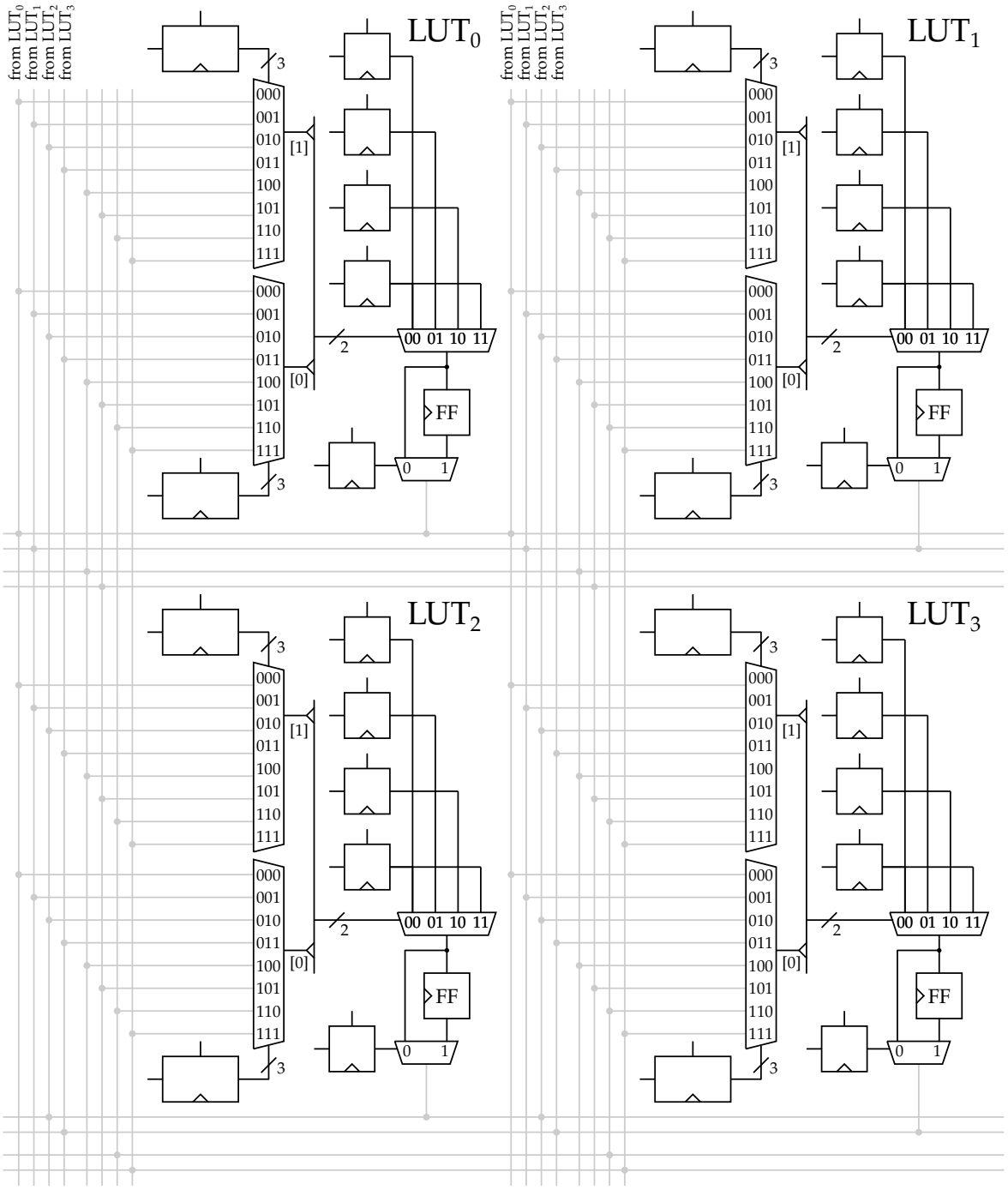
Which multiplier implementation would you choose in which situation?

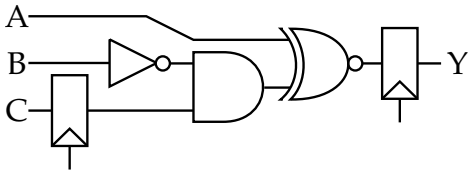
Problem 2. Map Logic to the FPGA



Part 2.A

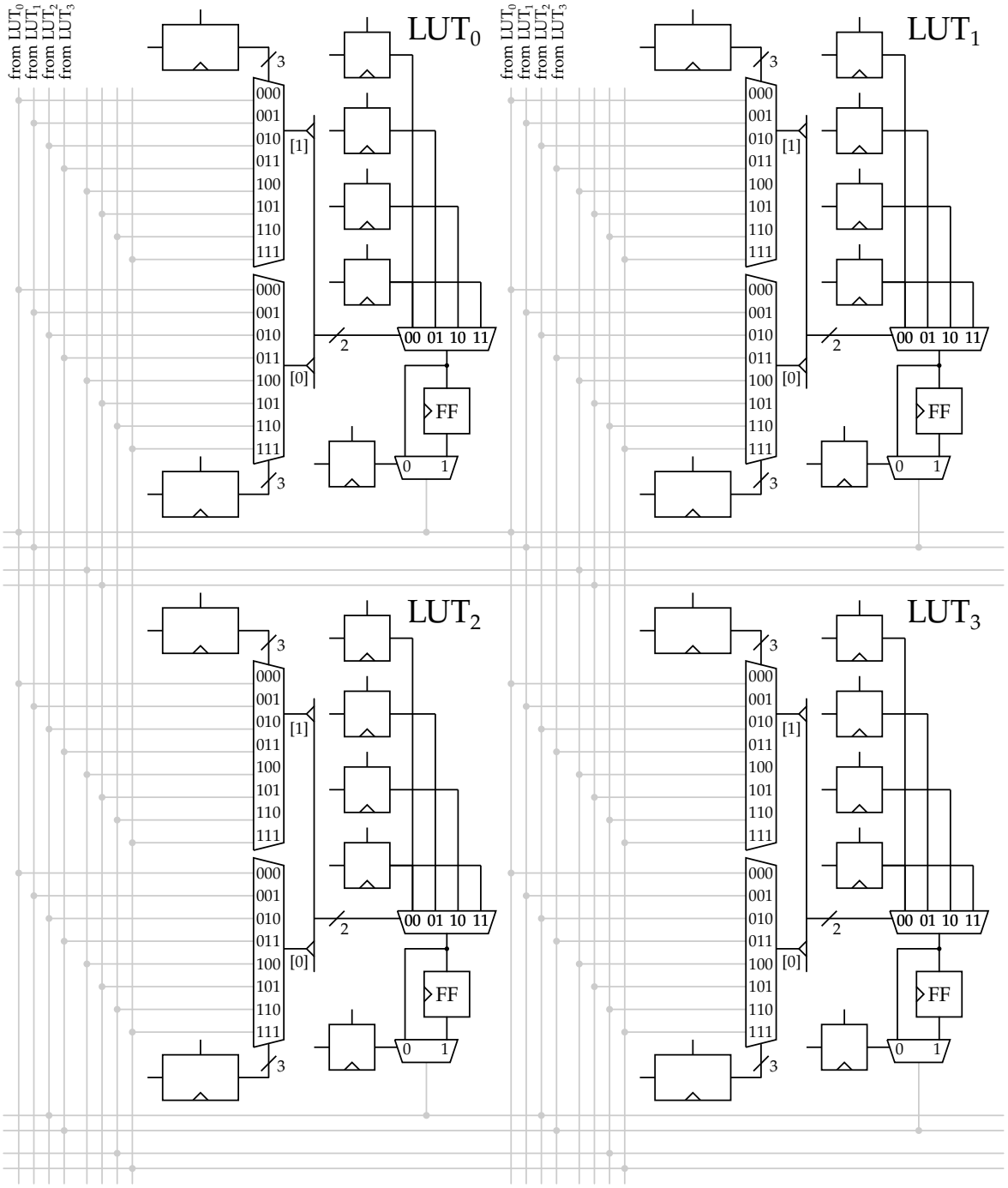
Map a NAND4 gate to the FPGA logic. Consider a tree structure.

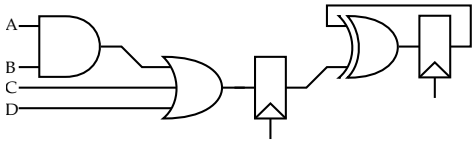




Part 2.B

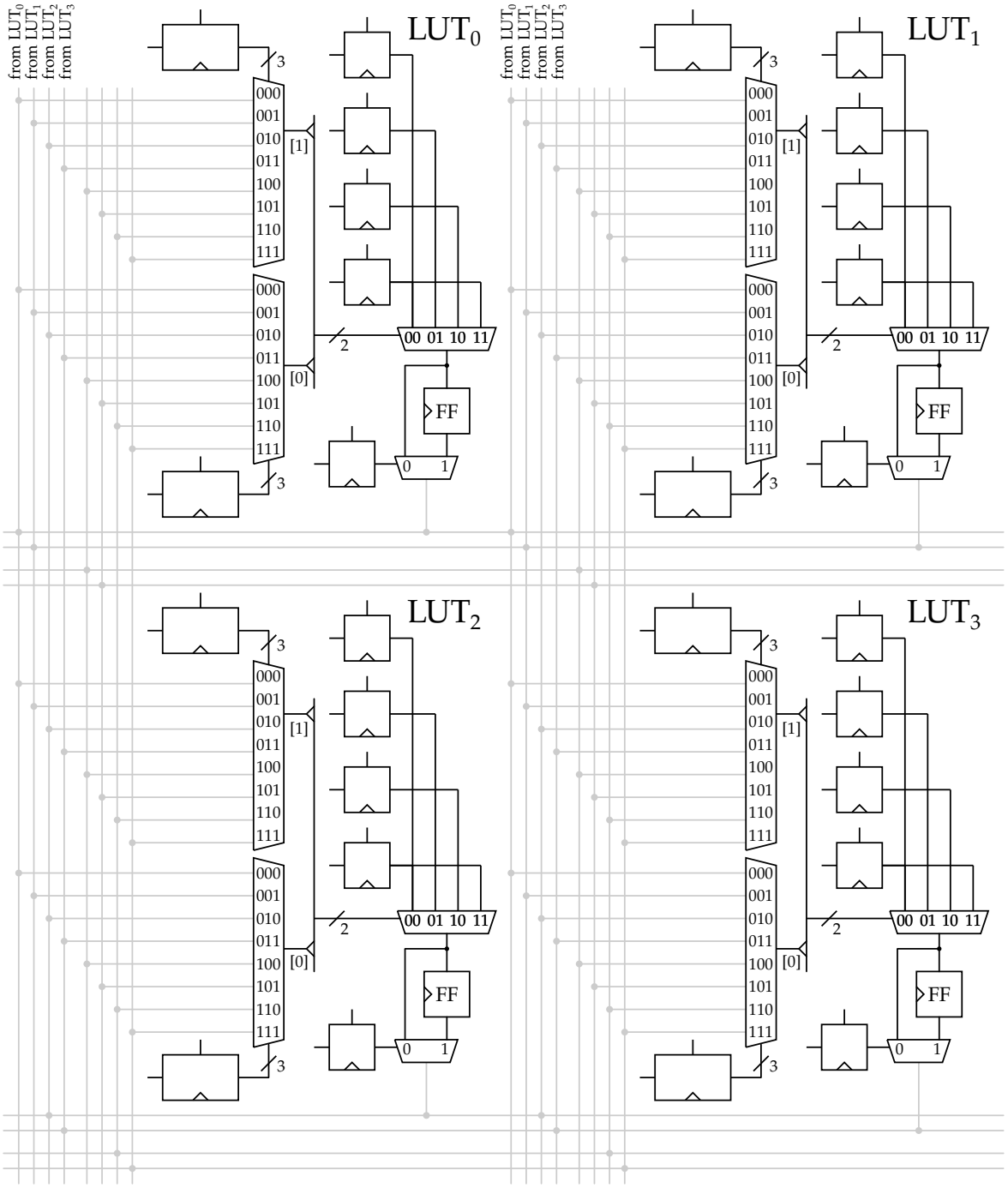
Map the gate network on the right to the FPGA.





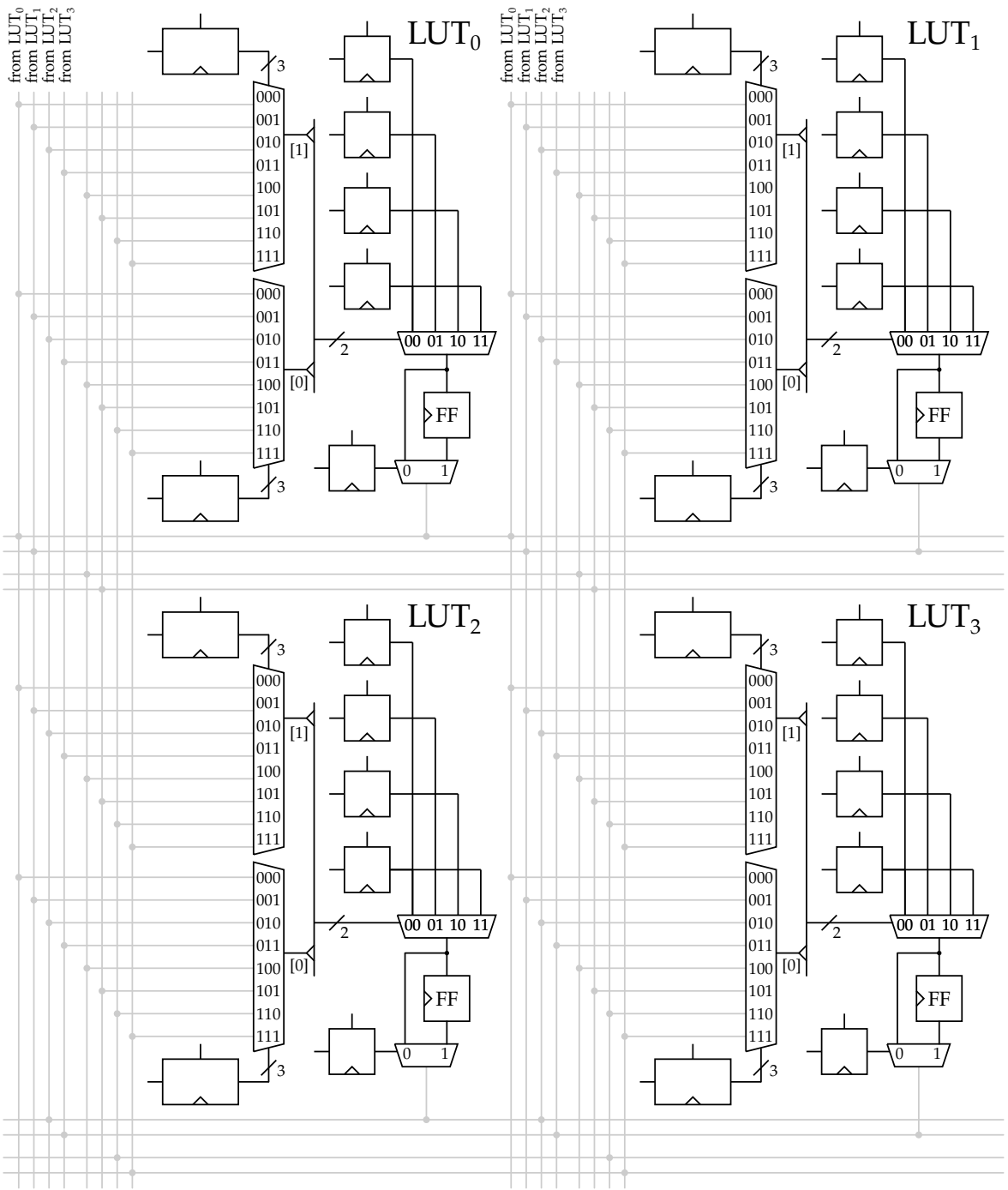
Part 2.C

Map the gate network on the right to the FPGA.



Part 2.D

Design a two bit equality comparator unit. Draw its gate network and map the network to the FPGA.



Part 2.E

Design a two bit incrementor with two half adders. Draw its gate network and map it to the FPGA.