HLS ASSIGNMENT — SOLUTION KEY

Total Points: 50 | Questions: 10 | Date: March 04, 2026

INSTRUCTOR COPY — CONTAINS ANSWERS & RUBRICS — DO NOT DISTRIBUTE TO STUDENTS

AI-generated graduate-level electrical assignment. Contains 10 questions covering key concepts.

Question 1

5.0 points

Explain the role of Control and Data Flow Graphs (CDFGs) in the high-level synthesis of FPGAs. How do CDFGs assist in optimizing hardware design?

ANSWER

CDFGs are used in high-level synthesis to model both control flow and data dependencies. By representing the design at an abstract level, they allow optimization of control paths and data routes, enabling efficient hardware mapping and resource allocation.

GRADING RUBRIC

- Full Marks: Clear explanation of what CDFGs are and their role in optimizing control and data paths in FPGA design. - Partial Marks: Description of CDFGs without clear explanation of optimization benefits. - No Marks: Incorrect or unrelated information.

Question 2

5.0 points

Given the task graph in the diagram below with three tasks T1, T2, and T3 executed on an FPGA, where T1 takes 5 ns, T2 takes 10 ns, and T3 takes 7 ns, and considering that T1 must finish before T2 and T3 can't start until T2 is done, what is the minimum execution time for all tasks on the FPGA?

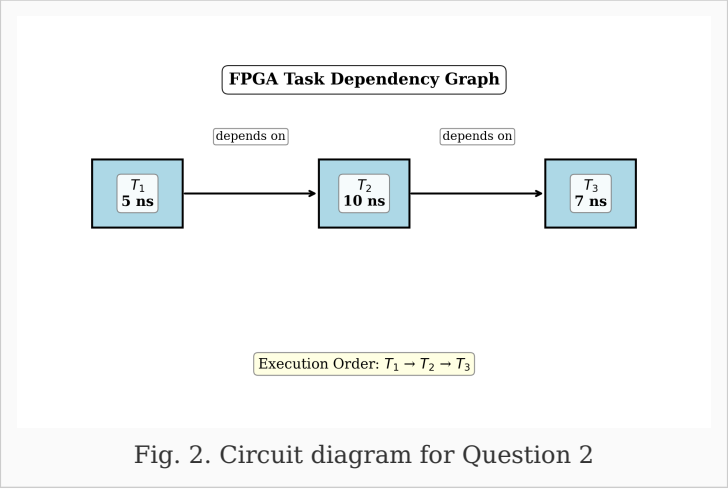

Fig. 2. Circuit diagram for Question 2

ANSWER

22 ns

GRADING RUBRIC

- Full Marks: Correct calculation leading to a minimum execution time of 22 ns. - Partial Marks: Correct method but calculation error. - No Marks: Incorrect method or completely incorrect answer.

Question 3

6.0 points

Analyze the task graph in the diagram below and determine the critical path. Assume each task's execution time is shown in the diagram. What is the length of the critical path?

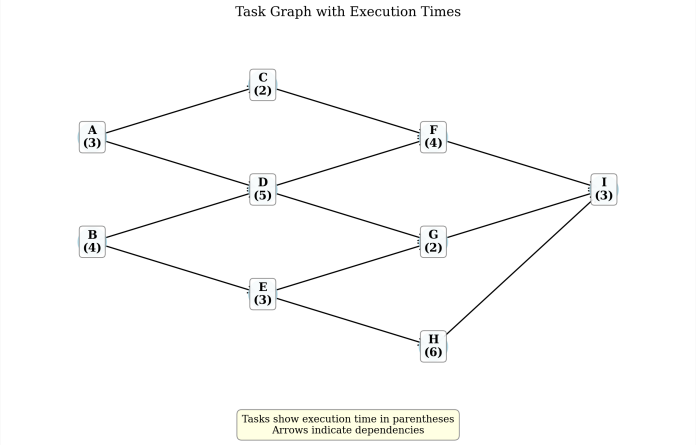

Fig. 3. Circuit diagram for Question 3

ANSWER

The critical path is the longest path through the task graph. Based on the diagram, it is T1 -> T3 -> T5 with a total time of 30 ns.

GRADING RUBRIC

- Full Marks: Correct critical path identification and calculation. - Partial Marks: Correct path but incorrect calculation. - No Marks: Incorrect path identification.

Question 4 6.0 points

Write a C++ function that represents a simple task scheduling algorithm for the task dependencies illustrated in the diagram below. Tasks: Task A must finish before Task B and Task C. Task B and C are independent after A finishes. Provide a function template.

```
void scheduleTasks() {
    // Define logic here
}
```

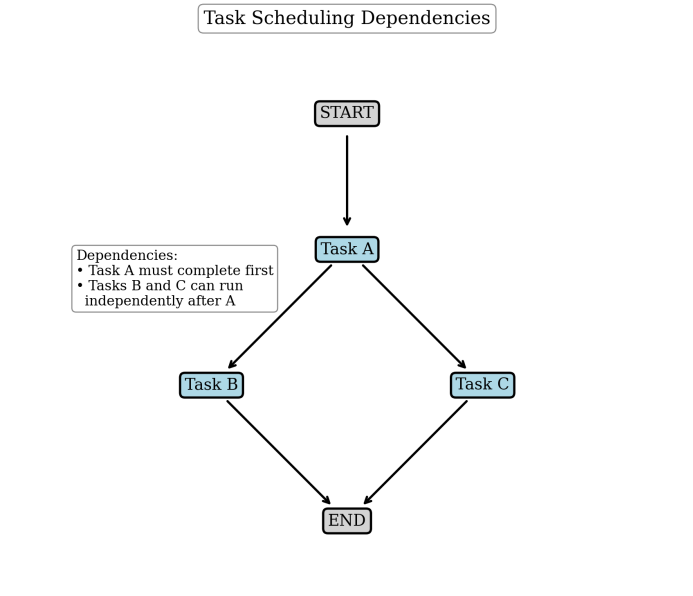

Fig. 4. Circuit diagram for Question 4

ANSWER

void scheduleTasks() { executeTask("A"); executeTask("B"); executeTask("C"); } void executeTask(const std::string &task) { // Code to execute the task }

GRADING RUBRIC

- Full Marks: Correct implementation of task scheduling in function. - Partial Marks: Correct logic but function syntax errors. - No Marks: Incorrect function logic or unrelated code.

Question 5

6.0 points

Convert the CDFG expressed in pseudo code from the diagram below to a set of RTL operations suitable for FPGA synthesis. Assume simple arithmetic operations take 1 cycle each. Pseudo Code: X = A + B; Y = X - C; Z = Y * D.

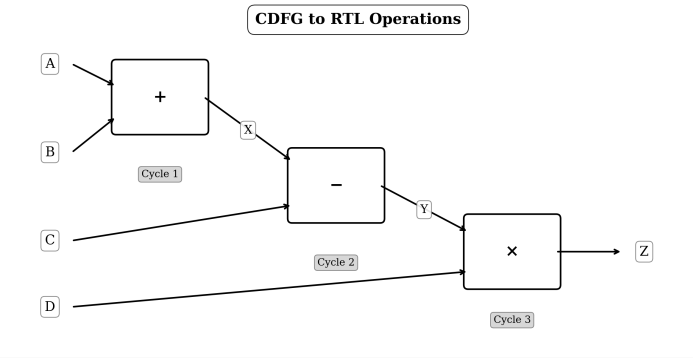

Fig. 5. Circuit diagram for Question 5

ANSWER

1. t1 = A + B; // Cycle 1 2. t2 = t1 - C; // Cycle 2 3. Z = t2 * D; // Cycle 3

GRADING RUBRIC

- Full Marks: Correct cycles and operation order. - Partial Marks: Correct operations, incorrect cycle assignment. - No Marks: Incorrect operations or cycle logic.

Question 6

5.0 points

Discuss the significance of task parallelism in FPGA-based systems compared to CPU-based systems. Focus on how high-level synthesis facilitates this parallelism.

ANSWER

Task parallelism in FPGA systems allows multiple processes to run simultaneously, leveraging hardware resources efficiently. High-level synthesis translates high-level parallel constructs into parallel hardware circuits, unlike sequential CPU execution.

GRADING RUBRIC

- Full Marks: Comparison of task parallelism between FPGA and CPU systems with emphasis on HLS. - Partial Marks: Discuss parallelism but lack HLS relevance. - No Marks: Incorrect or unrelated explanation.

Question 7 7.0 points

Given a C++ code snippet that calculates the greatest common divisor (GCD) of two numbers, convert this code into an equivalent Verilog module. Provide a Verilog code template for the GCD calculation.

```
module gcd(input wire [7:0] a, input wire [7:0] b, output wire [7:0]
gcd_out);
    // Declare internal signals
    // GCD logic
endmodule
```

ANSWER

module gcd(input wire [7:0] a, input wire [7:0] b, output wire [7:0] gcd_{out}); reg [7:0] x, y; always @(*) begin x = a; y = b; while (x != y) begin if (x > y) x = x - y; else y = y - x; end gcd_{out} = x; end endmodule

GRADING RUBRIC

- Full Marks: Correct conversion of GCD algorithm from C++ to Verilog, properly using always block. - Partial Marks: Key parts correctly converted but with syntax errors or logic errors. - No Marks: Unrelated code or incorrect logic.

Question 8

4.0 points

Examine the sample CDFG provided in the diagram below. Identify and list all dependencies between operations and categorize each as data or control dependencies.

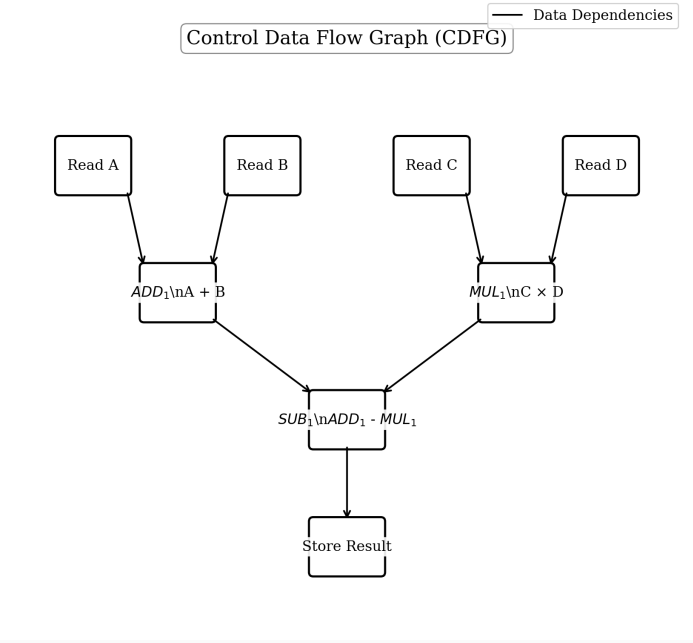

Fig. 8. Circuit diagram for Question 8

ANSWER

Dependencies: - Data: A -> B, B -> C - Control: D -> E

GRADING RUBRIC

- Full Marks: Correct identification and categorization of all dependencies. - Partial Marks: Some correct identifications but missed dependencies. - No Marks: Incorrect or missing information.

Question 9

3.0 points

Describe the process of converting a task graph into a hardware schedule suitable for FPGA implementation. Mention the challenges involved during this conversion.

ANSWER

Conversion involves mapping task nodes to functional units, determining execution order, and optimizing for resource and timing constraints. Challenges include managing resource conflicts and ensuring latency and throughput requirements are met.

GRADING RUBRIC

- Full Marks: Clear description of conversion process and challenges. - Partial Marks: Description of process without challenges. - No Marks: Incorrect or vague process description.

Question 10 3.0 points

Assume an FPGA implementation schedule must meet a throughput of 100 operations per microsecond. If each operation requires 100 ns on average, calculate the required level of parallelism (number of concurrent operations).

ANSWER

10 operations need to be executed in parallel.

GRADING RUBRIC

- Full Marks: Correct calculation of required parallel operations. - Partial Marks: Correct understanding but arithmetic error. - No Marks: Incorrect understanding or solution approach.