IIT JODHPUR

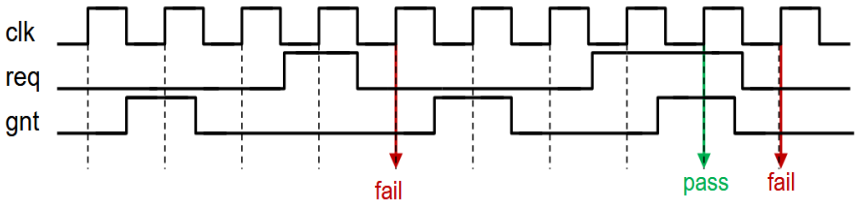
Major Examination: EEL1570 Hardware Software Co-design-Nov'25 (OPEN-BOOK)

Guidelines (Total time: 180 minutes, Maximum Marks: 45):

- Please read the question paper very carefully. **NO clarification is required in any question.** In case of any doubt, assume whatever you wish to and state that clearly in your answer.

---

1. Suppose the company *BharatSemi* where you had interned earlier decided to interview you for a full-time role. You are provided with the following debug dump where the assertion fail/pass scenarios are shown. You are asked to write the assertion (preferably SVA format) that can capture this dump. [2]



2. A very simple way to detect edges in an image (represented as $I(x, y)$) is through Laplacian filtering (i.e., application of laplacian operator as shown in uppermost left part of below figure). The other parts of below figure represent the discrete form of the laplacian operator (L) and eventually the convolution relationship (the equation mentioned in lower-half of the below figure) that shows how the output image ($I_L(i, j)$) can be obtained from the input image, $I(i, j)$.

$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$	**Discrete Laplacian kernel** $L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	**Laplacian-filtered image is simply the convolution:** $I_L(x, y) = (L * I)(x, y)$
Resultant Final Equation (Discrete form of convolution represented above): $I_L[i, j] = 4I[i, j] - I[i-1, j] - I[i+1, j] - I[i, j-1] - I[i, j+1]$		

With relevant assumptions (i.e., no. of clock cycles for individual operations etc.,), compare the full-hardware v/s full-software implementation if our goal is to achieve maximum throughput? [4]

3. A deep-tech startup called as *CipherTech* is involved in the development of encryption IP for usage in different security applications. To optimize the performance, the company decides to develop a hardware IP. The proprietary encryption algorithm involves the below function:

```
unsigned long int_sqrt(unsigned long n) {
    if (n == 0) return 0;
    unsigned long x = n;
    unsigned long next_x;
    while (1) {
        next_x = (x + n / x) / 2;

        // Stop when the value no longer improves
        if (next_x >= x)
            return x;

        x = next_x;
    }
}
```

Develop a fully synthesizable (and optimized) hardware block for the above function so that this block can be incorporated into the encryption IP to be launched by *CipherTech*? [5]

4. Consider a system where an ARM Cortex-M processor must interface with three major memories and two peripherals through an AHB interconnect and an APB bridge. The total system address map spans from 0x4000_0000 to 0xA000_0000. The system contains: External DRAM requiring 0.75 GB, On-chip SRAM requiring 0.25 GB, Boot ROM requiring 0.25 GB and General-purpose peripherals (GPIO, UART, SPI) require the remaining address space. Within the GPIO segment, a temperature sensor (connected through APB) is attached that requires 32 KB register space.

   - Compute the address boundaries (start and end addresses) for: Boot ROM, On-chip SRAM, External DRAM, Peripherals and Temperature sensor (inside APB space). Ensure that the assigned address ranges exactly fill the total address span without overlap. [3]
   - Draw and briefly describe the complete AHB/APB system interconnection diagram for connecting all the above components? [2.5]
   - Explain briefly how data from the temperature sensor is fetched by the ARM CPU through the AHB/APB hierarchy? [2.5]

5. You have been hired by an IITJ-incubated startup to design an object-detection inference engine for an edge AI-enabled smart camera. The hardware inventory available to you contains:

   - A baseline CPU with the following characteristics:
     - Clock frequency: 1.6 GHz
     - Average CPI (Cycle per Instruction) for integer instructions: 1.0
     - Average CPI (Cycle per Instruction) for floating-point instructions: 2.5
     - Dynamic power when executing integer instructions: 1.2 W
     - Dynamic power when executing floating-point instructions: 2.4 W
     - Static (leakage) power: 0.4 W
     - Area footprint: 0.25 mm^2
   - An ASIC block (can be placed within the same area as the CPU) that contains 4 dedicated ALUs (as defined below):
     - ALU-FP1: scalar FP unit — latency 3 cycles/op, energy 8 nJ/op
     - ALU-FP2: vector FP unit — latency 2 cycles/op (when fed with packed data), energy 6 nJ/op
     - ALU-INT1: integer ALU — latency 1 cycle/op, energy 1.5 nJ/op
     - ALU-INT2: integer ALU — latency 1 cycle/op, energy 1.5 nJ/op
     - ASIC block's static power = 0.3 W when enabled

   You must decide between (i) using the CPU only, (ii) using the ASIC only (offloading all data-related instructions to the ASIC), or (iii) a hardware–software co-design in which the compiler/a runtime controller maps a fraction of instructions to the ASIC ALUs and the rest run on the CPU. The ASIC and CPU can operate concurrently; assume the CPU continues to execute non-offloaded instructions while the ASIC executes offloaded instructions. The object-detection executable produced by your toolchain has 12,000 instructions in total. Instruction mix among these 12,000 instructions is: Floating-point (FP) instructions: 35% and Integer (INT) instructions: 65%.
   Which option you would choose to implement if goal is to maximize performance and minimize power consumption? Show your calculations with relevant assumptions. [6]

   Hint: For the CPU-only execution, convert per-instruction CPI and clock to execution time and compute energy using (dynamic power) × (execution time) + (static power) × (total time).

6. Suppose you are designing a hardware accelerator for a lightweight cryptographic algorithm to be deployed inside an IoT security module by the company *TharIoTSystems*. One crucial step in this algorithm is the generation of a sequence of GCD values between incoming data words and a fixed secret key parameter. To ensure high security and minimum host-CPU involvement, the GCD computation must be done entirely on-chip. You are told that a continuous stream of 32-bit integers arrives at your accelerator input port. For each incoming number N, your hardware must:

   - Compute GCD(N,K) where K is 32-bit key stored in a secure register.
   - Assert a signal valid_out and output the GCD result.
   - Repeat the process continuously for all values arriving in the stream.

   You must design the hardware unit that performs the computation using least resources and the computation gets executed in almost real-time. **Design the optimized datapath and controlpath for this hardware such that latency is minimized. Further, taking hints (whichever is applicable here) from values provided in previous question, find the power consumption and system latency designed by you for** *TharIoTSystems***?** [6]

7. The company *SmartNetworkDesign* develops custom network traffic (measured in the unit of packets) controller that performs packet length accumulation and generates alerts when specific thresholds are reached. You have joined this company for an internship role with the condition that your performance during internship may lead to a full-time role.

```
module packet_counter(
   input  wire        clk,
   input  wire        rst,
   input  wire        pkt_valid,
   input  wire [7:0]  pkt_len,
   output reg  [15:0] total_len,
   output reg         warn,
   output reg         overflow
);
   always @(posedge clk) begin
      if (rst) begin
         total_len <= 0;
         warn      <= 0;
         overflow  <= 0;
      end else if (pkt_valid) begin
         total_len <= total_len + pkt_len;

         if (total_len + pkt_len > 16'h0FFF)
            warn <= 1;
         else
            warn <= 0;

         if (total_len + pkt_len > 16'hFF00)
            overflow <= 1;
      end
   end
endmodule
```

   You are asked to develop a complete HDL (e.g. Verilog/SystemVerilog) testbench for this design with the explicit goal of maximizing toggle coverage for the signals of the design (i.e., all internal registers, input signals and conditions inside procedural blocks). Is this testbench sufficient enough for the verification of the above design? [4+2]

   Hint: Toggle coverage maximization means the variables change from 0 to 1 or 1 to 0.

8. Due to a dull core job market in this placement season at IIT Jodhpur, instead of accepting non-core job offers, you decide to incubate a startup with the support of a few seniors and college professors. You decide to name your company as *DrishtiVisionSystems* and the first customer of your company is the state government that has tasked you to develop a real-time video analytics system for a street surveillance device. The system must process a continuous stream of frames at 60 frames per second (FPS), performing the following tasks on each frame:

- i) Frame acquisition from the camera,
- ii) Noise filtering (pixel-wise smoothing),
- iii) Edge detection,
- iv) Object tagging and classification using a ML model

You are provided with the following constraints and resources:

- The embedded processor can execute 200 million instructions per second (200 MIPS).
- A custom hardware accelerator (FPGA/ASIC) can be used, with a budget of 40k logic elements and 20 DSP slices.
- The end-to-end frame processing budget is 16 ms per frame.
- Software execution time estimates (if implemented entirely on CPU):
  - Noise filtering: 6 ms per frame
  - Edge detection: 8 ms per frame
  - Object classification: 12 ms per frame
- Hardware implementation estimates are as follows:
  - Noise filtering in HW: 1.2k logic + 4 DSP, latency 1 ms
  - Edge detection in HW: 3k logic + 6 DSP, latency 1.5 ms
  - Object classifier in HW: 30k logic + 16 DSP, latency 3 ms

Propose a hardware–software partitioning for the system such that the 60 FPS real-time constraint is met. Identify which tasks should be mapped to hardware and which should remain in software. Please provide justification behind your choice using timing and resource arguments. [5]
Based on your solution to the above question, please estimate the total frame processing time after partitioning and verify whether the design satisfies the 16 ms/frame limit. [3]

Hint: A generalized representation of this system is shown as below: