Grading scale: 00–25: insufficient 26–31: sufficient 32–38: satisfactory
39–44: good 45–50: very good

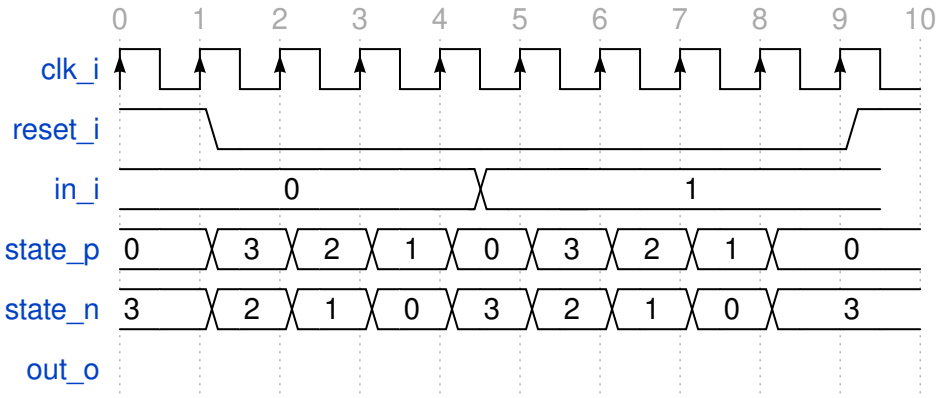
The use of examination aids (e.g., calculators) is prohibited. Answers can be given in German or English. Please refrain from using lead pencils and red ink pens.

Some questions specify an approximate sentence count for your answer. This should give you an idea of how much detail is expected, and is *not* a hard limit. Avoid giving a two-paragraph answer to a question with "approx. 2 sentences."

Matr. number: **Last name:**

---

1. (10 points) **Finite State Machine:** Consider the FSM provided in the timing diagram below. The three top-most signals are 1-bit signals. The three signals at the bottom are 2-bit signals. in_i is an input signal for this FSM and state_p is the current state. state_n and out_o are outputs of the next state logic and output logic respectively.



   (a) Give the two logical formulas of the next state logic.

   (b) The output logic is defined as below. Add the missing out_o line in the timing diagram (reminder: LSB is at index 0).

$$\text{out\_o}[0] = (\text{state\_p}[0] \vee \text{state\_p}[1]) \vee \neg\text{in\_i}$$
$$\text{out\_o}[1] = \text{state\_p}[0] \wedge \neg\text{state\_p}[1]$$

   (c) Draw the ASM diagram of this machine.

   (d) Which type of Finite State Machine is this FSM? What is the name of the other type? What is their difference?

→

2. (10 points) **Pipelining:**
   (a) Explain the motivation and concept of pipelining based on the following example. You have a single cycle machine with a critical path of 600ps. You are able split the path into three pipeline stages of 200ps each. What does it mean to insert a pipeline stage? Explain how this leads to a speedup. How much is the speedup (assuming a full and ideal pipeline)?
   (b) Assume the three pipeline stages `IF`, `ID` (includes reading the operands from the register file), and `EX`. This pipeline executes the three instructions below. Explain the problem arising in the pipeline with this code snippet. Explain two approaches to resolve this problem.
   (c) Complete the diagrams below for both approaches and show which pipeline operation is performed at what clock cycle for each instruction of the code snippet. Assume an empty pipeline before starting the execution. State the total number of needed clock cycles for the execution of the code snippet for both approaches.
   (d) Explain the concept of *Scoreboarding* and discuss why its usage for pipelining.

```
ADDI x2, x1, 0  # I1
ADDI x3, x1, 5  # I2
ADDI x4, x3, 3  # I3
```

→

3. (10 points) **Assembly:**
   (a) A RISC-V CPU fetches the 32-bit instruction with the value `0x0201A583`. What instruction does the CPU process? Decode all fields of the instruction.
   (b) Transform the following C-code to RISC-V assembly. All local variables of the C-code **must** be allocated on the stack. The global variable `g` is located at address `0x7F0`. The RISC-V calling convention must be followed. The assembly startup code including the initialization of the stack is provided below. Write the assembly code for the two functions at the foreseen locations.

Assembly Reference

31 26	25 20	19 15	14 12	11 7	6 0	
imm[11:0]		rs1	010	rd	0000011	`LW rd,imm(rs1)`
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	`SW rs2,imm(rs1)`
0000000	rs2	rs1	000	rd	0110011	`ADD rd,rs1,rs2`
imm[11:0]		rs1	000	rd	0010011	`ADDI rd,rs1,imm`
0100000	rs2	rs1	000	rd	0110011	`SUB rd,rs1,rs2`

```c
// Located at memory address 0x7F0
int g;

int subf(int* p, int t) {
  return *p + t;
}

int main() {
  int a = 7;
  g = 2;
  return subf(&g, a) + a;
}
```

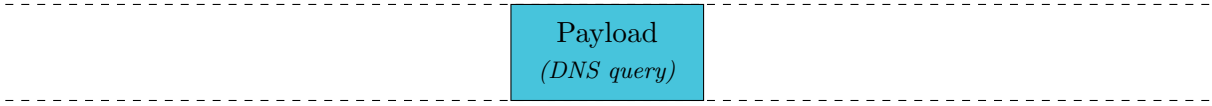
```
_start:
  ADDI sp, zero, 0x700
  JAL ra, main
  EBREAK

subf:
```

```
main:
```

→

4. (10 points) **Encapsulation:**
   (a) (3 points) Pictured below is a DNS query, which is being sent to some DNS resolver. As this DNS query goes down the protocol stack, each layer adds information. Complete the drawing so it depicts the resulting Ethernet frame's structure. *(You do not need to draw individual fields at each layer, only the general structure.)*
   (b) (4 points) Once this Ethernet frame arrives at the default gateway, describe how is it processed. What layers are looked at, in what order? What fields are modified, and how? Stop once the data leaves the default gateway. *(Use approx. 4-7 sentences.)*
   (c) (3 points) Assume that the default gateway also performs *Network Address Translation (NAT)*. What fields are modified as a result, and how? *(Use approx. 3-4 sentences.)*



5. (10 points) **TCP data stream control:**
   (a) (4 points) The *Transmission Control Protocol* guarantees data ordering and reliable delivery. Explain how it does so. What header fields are used? How does this differ for data being sent by the server or the client respectively? *(Use approx. 3-5 sentences.)*
   (b) (4 points) In the lecture, we discussed two separate concerns for transmission speed. Describe and contrast *flow control* and *congestion control* in TCP. How does is each mechanism work, and what is it trying to avoid? *(Use approx. 2-3 sentences each.)*
   (c) (2 points) If a data packet in a TCP connection is lost, this can lead to inefficient use of bandwidth. Explain the concept of *selective acknowledgements. (Use approx. 2-3 sentences.)*

Good Luck!