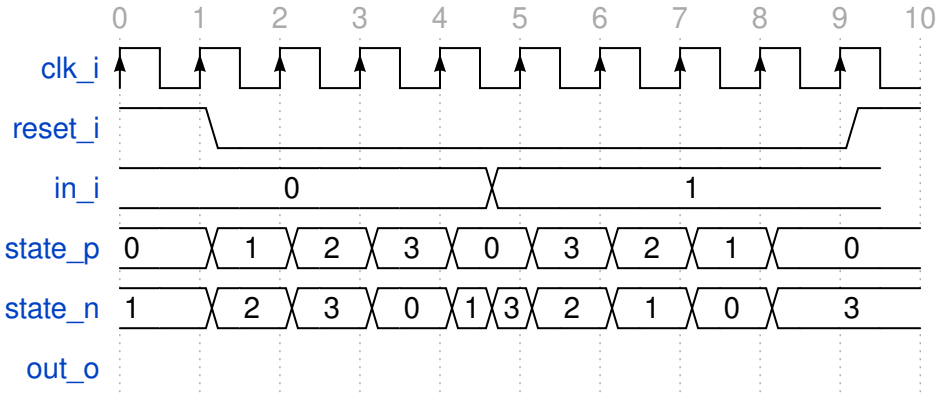
Grading scale: 00–25: insufficient 26–31: sufficient 32–38: satisfactory
39–44: good 45–50: very good

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.

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 logical formulas for both bits of the next state logic.
   (b) The output logic is defined in Verilog as below. Add the missing out_o line in the timing diagram (reminder: LSB is at index 0).

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

   (c) Draw the ASM diagram of this machine.
   (d) Name the type of this Finite State Machine. What is the name of the other type? What is their difference?

2. (10 points) **Memory and Cache:** Assume a directly-mapped data cache with a total size of 64 bytes, organized in 8 blocks, and 256 bytes of byte-addressable main memory.
   (a) What is a memory hierarchy and why do we need one?
   (b) Name and explain the two types of locality that caches exploit.
   (c) How many bits are needed for addressing the main memory and how many bits of the address are used for tag/index respectively?
   (d) Sketch the directly-mapped cache and explain how a cache access to the address 0x56 is performed. What checks are performed on which data?
   (e) What is a replacement policy in the context of caches? Name **2** examples.

3. (10 points) **Assembly:**
   (a) What is a calling convention and why is it needed? Explain what a calling convention covers.

→

(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 `0xF00`. 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.

(c) Draw the state of the stack (memory cells with values and annotation, what is contained) between the lines `g = 4;` and `return addfunc(&a);`. Draw the state of the stack before executing `return *p + g;`

```c
// Located at memory address 0xF00
int g;

int addfunc(int* p) {
  return *p + g;
}

int main() {
  int a = 3;
  g = 4;
  return addfunc(&a);
}
```

Assembly Reference

```
LW    rd,imm(rs1)
SW    rs1,imm(rs2)
ADD   rd,rs1,rs2
ADDI  rd,rs1,imm
SUB   rd,rs1,rs2
JAL   rd,imm
JALR  rd,imm(rs1)
```

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

  addfunc:
```

```
main:
```

4. (10 points) **Data link and network layer:**
   (a) What does ARP stand for and how many bytes does a MAC address have? [2 pts]
   (b) What is ARP's purpose? [2 pts]
   (c) What is a gratuitous ARP message? [1 pt] When is it sent? [1 pt]
   (d) What is SLAAC's purpose? How does it work? [2 pts]
   (e) Which privacy aspects have to be considered with it? How are they prevented? [2 pts]

5. (10 points) **TCP/IP:**
   (a) What is the purpose of flow control? [1 pt]
   (b) Acknowledging each packet is slow. How can data throughput be increased? [2 pts]
   (c) What is the idea of sliding windows? Who specifies the advertised window? [2 pt]
   (d) Illustrate the principle of sliding windows with associated protocol parameters. Illustrate flow control by changing the parameters. [3 pts]
   (e) What is the difference between flow and congestion control? [2 pts]

Good Luck!