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 Automaton:** Given the following truth table of a synchronous automaton consisting of two flip flops (s1, s0), a one-bit input (in), and a two-bit output (out1, out0):
   (a) Show the corresponding ASM diagram of the automaton.
   (b) Show the structural diagram of the automaton featuring logic blocks, flip flops, and wires.
   (c) Specify the logical formulas for the individual logic blocks.
   (d) Name the type of automaton in this example. What is the name of the second type of automaton and explain the difference between them.
   (e) Change exactly one line of the truth table to create the other type of FSM. The resulting automaton need not be functionally equivalent.

s1	s0	in	out1	out0	next_s1	next_s0
0	0	0	0	0	0	1
0	0	1	0	0	0	1
0	1	0	0	1	1	0
0	1	1	0	1	1	1
1	0	0	1	0	0	1
1	0	1	1	1	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	1

2. (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 annotation, what is contained) between the lines `g = 4;` and `return addfunc(&a);` as well as 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:
```

3. (10 points) **Memory and Cache:**

   Assume a directly-mapped data cache with a total size of 32 bytes, organized in 4 blocks, and 128 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) What is the advantage/disadvantage of a set-associative cache over a directly mapped cache?
   (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 and what are the expected values for a cache hit?
   (e) What is a replacement policy in the context of caches? Name **2** examples.

4. (10 points) **IPv6:**

   (a) How many bits does an IPv4 address have? Equivalently, for IPv6? [1 pt]
   (b) Mention the rules to simplify IPv6 addresses. Considering `2001:0db8:0:0:8a3:0:0:0`, give the shortest equivalent representation by the number of characters [3 pts]
   (c) Define the three address types Unicast, Multicast, and Anycast [3 pts]
   (d) Why is SLAAC (Stateless Address Auto-Configuratoin) stateless? [1 pt]
   (e) Which privacy aspects have to be considered with SLAAC? How to improve? [2 pt]

5. (10 points) **HTTP & DNS:**

   (a) Explain idempotence in the context of HTTP. Is HTTP OPTIONS idempotent? [2 pts]
   (b) Name at least 3 new features introduced by HTTP 2.0 [3 pts]
   (c) DNS replies are built up of resource records. Give example values for any 2 resource records (hint: `A`, `AAAA`, `CNAME`, `MX`, ...) [2 pts]
   (d) DNS lookups at authoritative servers take long. How is infrastructure sped up? [2 pts]
   (e) Facebook was unaccessible this week. Give one potential network-level reason [1 pt]

Good Luck!