Homework Assignment

1. Which of the following statements is NOT correct about zero-knowledge proofs?
   - ○ All NP languages have zero-knowledge interactive proofs.
   - ○ Interactive Proofs can efficiently verify any language in PSPACE.
   - ○ Single-prover Interactive Proofs can prove all languages that can be proven by Multi-prover Interactive Proofs.
   - ○ ZK proof of knowledge further proves that the prover knows a witness.

2. Which of the following statement is NOT correct about the applications of zero-knowledge proofs.
   - ○ zkRollup relies on the zero knowledge property to improve the scalability of blockchains.
   - ○ ZKP can be used to prove that a private transaction is valid on a public blockchain.
   - ○ One can verify that post-processed photos are taken by a camera by verifying the digital signature and the post-processing in zero-knowledge proofs.
   - ○ Zero-knowledge proof can be used to replace password to prevent identity theft.

3. Which of the ZKP schemes is based on bilinear pairing?
   - ○ Bulletproofs
   - ○ Plonk
   - ○ Interactive proofs
   - ○ Stark and FRI

4. Which of the ZKP schemes does not rely on Fiat-Shamir to be non-interactive?
   - ○ Plonk
   - ○ Stark and FRI
   - ○ Brakedown and Orion based on error-correcting code
   - ○ Groth16 based on linear PCP

5. Which of the following statements is correct about polynomial commitments?

   ○ The security of the KZG polynomial commitment is only based on the q-SBDH assumption.

   ○ The polynomial commitment based on Bulletproofs has a logarithmic verifier time.

   ○ We can construct polynomial commitments based on error-correcting codes that do not have an efficient decoding algorithm

   ○ The polynomial commitment based on FRI has a square-root proof size.

6. Which of the following is correct about the sumcheck protocol?

   ○ The randomness selected by the verifier has to be kept secret from the prover.

   ○ The randomness selected by the verifier depends on the message sent by the prover in the previous round.

   ○ The proof size is logarithmic in the size of the multivariate polynomial.

   ○ The verifier time is logarithmic in the size of the multivariate polynomial.

7. Which of the following statements is NOT correct about Plonk?

   ○ Plonk relies on the Schwartz-Zippel lemma to prove polynomial equations.

   ○ The vanishing polynomial of a set evaluates to 0 at all points in the set.

   ○ Plonk protocol can support circuits with gates other than addition and multiplication.

   ○ Only KZG polynomial commitments can be used to compile Plonk-IOP to a ZKP scheme.

8. Which of the following statement is correct about IVC?

   ○ SNARKs are necessary to realize IVC.

   ○ The verifier complexity of IVC from folding scheme is smaller than that of IVC from succinct verification.

   ○ The recursion overhead in IVC from folding scheme is smaller than that in IVC from succinct verification.

   ○ The prover complexity of IVC from folding scheme is smaller than that of IVC from succinct verification.

9. Which combination has the fastest prover?

   ○ Sumcheck IOP (from lecture 4) + Bulletproof PC

   ○ Sumcheck IOP (from lecture 4) + Orion PC

   ○ Plonk IOP (from lecture 5) + KZG

○ Plonk IOP (from lecture 5) + FRI PC

10. Which combination has the shortest proof size?

    ○ Sumcheck IOP (from lecture 4) + Bulletproof PC

    ○ Sumcheck IOP (from lecture 4) + Orion PC

    ○ Plonk IOP (from lecture 5) + KZG

    ○ Plonk IOP (from lecture 5) + FRI PC

11. *[45 points]* One of the more challenging notions to wrap one's head around regarding the interactive proof protocol in Lecture 4 is that, when applying it to a circuit C with a "nice" wiring pattern, the verifier never needs to materialize the full circuit. This is because the only information about the circuit's wiring pattern of C that the verifier needs to know in order to run the protocol is to evaluate $\widetilde{\mathsf{add}}$ and $\widetilde{\mathsf{mult}}$ at a random point, and $\widetilde{\mathsf{add}}$ and $\widetilde{\mathsf{mult}}$ often have nice, simple expressions that enable them to be evaluated at any point in time logarithmic in the size of C.

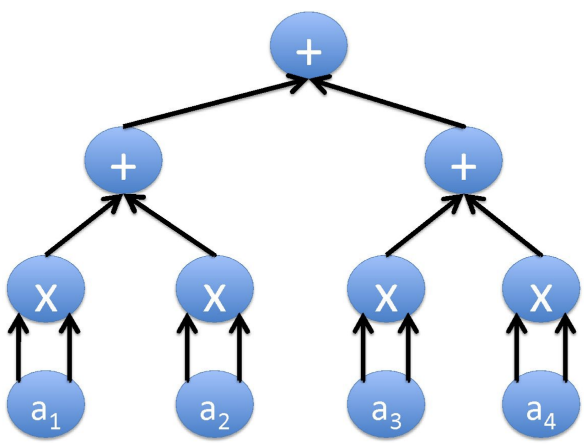

Figure 1: Q11 Circuit for input size $n = 4$.

This problem asks you to work through the details for a specific, especially simple, wiring pattern. Figure 1 depicts (for input size $n = 4$) a circuit that squares all of its inputs, and sums the results via a binary tree of addition gates.

(a) *[10 points]* Give an expression for the multilinear extension $\widetilde{\mathsf{eq}}_\ell$ of the equality predicate $\mathsf{eq}_\ell : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$, that is defined as follows:

$$\mathsf{eq}_\ell(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

Argue that this polynomial can be evaluated at any point in time $O(\ell)$.

(b) *[15 points]* Assume that n is a power of two. Give an expression for $\widetilde{\mathsf{mult}}$ that can be evaluated at any point in time $O(\log n)$. The multiplication layer consists of $n = 2^{d-1}$ multiplication gates, where the j-th multiplication gate at layer $d-1$ has both in-neighbors equal to the j-th input gate at layer d.

(c) *[20 points]* Assume that n is a power of 2. Give an expression for $\widetilde{\mathsf{add}}$ that can be evaluated at any point in time $O(\log n)$. The addition layer i consists of 2^i addition gates, where for $j \in \{0, 1, \ldots, 2^i - 1\}$, the j-th addition gate at layer i has as its in-neighbors gates $2j$ and $2j+1$ at layer $i+1$.

12. *[45 points]* (Custom gates in Plonk) Lecture 5 covers the Plonk protocol for arithmetic circuits with addition gates and multiplication gates, but the same technique generalizes to custom gates. In this question, you will extend the Plonk protocol to include a custom gate g with the following specification: $g(a, b, c) = 3a^4 + 7a^2bc - 2bc^2$. Note that it requires 11 addition/multiplication gates to evaluate one instance of g. Assume that the circuit C we want to prove has ℓ inputs, m addition/multiplication gates, and n instances of the custom g gate.

(a) *[10 points]* Define the following for the extended Plonk protocol (see Lecture 5 slides for reference):

- the set Ω (Slide 42)
- the set Ω_{inp} (Slide 47)
- the set Ω_{gates} (Slide 49)
- the trace polynomial T (Slide 43)
- the selector polynomial S (Slide 48)

(b) *[20 points]* Let d be the degree of the trace polynomial T. Assuming the wiring polynomial $W \in \mathbb{F}_p^{(\leq d)}[X]$ is already given, describe the checks required to ensure the validity of T (see Slide 53 from Lecture 5 for reference) in the extended Plonk protocol. For each check, specify the proof gadget (Slide 38), along with the degree of the polynomial and the size of the set on which the gadget is used.

(c) *[15 points]* Assuming the following:

- KZG (Lecture 6) is used to instantiate the polynomial commitment
- KZG evaluation proofs are only batched across queries to the same polynomial
- Irrespective of the number of queries to a committed polynomial f, the corresponding quotient polynomial q in the batched evaluation proof of KZG is committed with the same degree bound as f

Find the total number of group exponentiations performed by the prover to prove circuit C using the extended Plonk protocol. For comparison and as a reference, when proving C using the Plonk protocol (without custom gates) under the same assumptions, the total number of exponentiations performed are $40(11n+m)+11\ell$.