Midterm

Name:

1. **BDD package** [5 marks]

   What are the advantages (if any) and disadvantages (if any) of inserting the result of $ITE(f, g, h)$ into the cache when f is the constant function 0 or 1?

2. **Algebraic methods**

(a) (Factoring) [10 marks]

Here is an equation I read into SIS (using read_eqn):

```
f = ( a + b * ( c + d + a * ( x + y ) ) ) * ( d + q ( x + y ) );
```

When I asked SIS to factor it, it was able to produce a factored form with 9 literals.

Find the best possible factored form you can find for f. (You can eyeball it, or try and be systematic, e.g., divide out a kernel.)

(b) Given algebraic expressions for f and g:

$$\begin{aligned} f &= ab' + a'b + a'c' + b'c' + a'd + b'd \\ g &= a' + b' \end{aligned}$$

i. Report the number of literals in the expressions for f and g when f is expressed in terms of g using (1.) algebraic division, and (2.) Boolean division. (For the latter, you can just compute the quotient and remainder by inspection.) [10 marks]

3. **Boolean methods**

Consider the following logic equations:

```
f = a * b * c;
a = x * y;
b = y * z;
c = z * x;
```

(f is the only output, and x, y, z are the inputs.)

(a) SDC [5 marks]

When I run the *simplify* command (which simplifies logic with respect to satifiability don't cares) in SIS on these equations, I get no reduction. Why is that?

(b) ODC [10 marks]

i. (ODC) Explain why the ODCs for nodes a, b, c in terms of the primary inputs are $x' + y' + z'$ for each of the three nodes.

ii. Use ODC to simplify the node functions at a, b, c individually without considering the fact that the other nodes are being simplified.

A. What is the resulting network?

B. Why is it functionally different from the original network?

C. What would the right way be to use the ODCs for simplifying the network?

4. **Technology Mapping** [20 marks]

   Map the circuit in Figure 1 to two-input NAND gates and inverters, and determine an optimum covering using the library in Figure 2.

5. **Timing Analysis** [10 marks]

   Recall a path π in a combinational logic network is said to be statically sensitizable if it is not possible to find values for the primary inputs to the network that sets the side inputs on the gates on π to noncontrolling values.

   Explain what static sensitization is trying to do, and why it *fails* to achieve its goal.