Prime Number Identifier Circuit Testing Using Maxima

Author: Arsalan Khan 210862640
Date: 2023-10-27

Introduction:

This document outlines the testing procedure of the Prime Number Identifier Circuit using Maxima. The primary goal is to verify the correctness of the logic equation by running it through all possible binary input combinations from 0000 to 1111.

Equation Definitions:

```
t1: (not a3) and (not a2) and a1;
t2: (not a3) and a2 and a0;
t3: (not a2) and a1 and a0;
t4: a2 and (not a1) and a0;

prime: t1 or t2 or t3 or t4;
```

Testing Procedure:

The logic equation was inputted into the Maxima online calculator. Each binary combination was tested, and the outputs were documented.

Truth Table:

The given equation follows this truth table:

a3	**a2**	**a1**	**a0**	**Binary**	**Decimal**	**Prime?**
0	0	0	0	0	0	No
0	0	0	1	1	1	No
0	0	1	0	10	2	Yes
0	0	1	1	11	3	Yes
0	1	0	0	100	4	No
0	1	0	1	101	5	Yes
0	1	1	0	110	6	No
0	1	1	1	111	7	Yes
1	0	0	0	1000	8	No
1	0	0	1	1001	9	No
1	0	1	0	1010	10	No
1	0	1	1	1011	11	Yes
1	1	0	0	1100	12	No
1	1	0	1	1101	13	Yes
1	1	1	0	1110	14	No
1	1	1	1	1111	15	No

Maxima Session Code (input):

```
t1: (not a3) and (not a2) and a1;
t2: (not a3) and a2 and a0;
t3: (not a2) and a1 and a0;
t4: a2 and (not a1) and a0;
prime: t1 or t2 or t3 or t4;

/* Testing all 16 combinations */
prime, a0=false, a1=false, a2=false, a3=false;
prime, a0=true, a1=false, a2=false, a3=false;
prime, a0=false, a1=true, a2=false, a3=false;
prime, a0=true, a1=true, a2=false, a3=false;
prime, a0=false, a1=false, a2=true, a3=false;
prime, a0=true, a1=false, a2=true, a3=false;
prime, a0=false, a1=true, a2=true, a3=false;
prime, a0=true, a1=true, a2=true, a3=false;
prime, a0=false, a1=false, a2=false, a3=true;
prime, a0=true, a1=false, a2=false, a3=true;
prime, a0=false, a1=true, a2=false, a3=true;
prime, a0=true, a1=true, a2=false, a3=true;
prime, a0=false, a1=false, a2=true, a3=true;
prime, a0=true, a1=false, a2=true, a3=true;
prime, a0=false, a1=true, a2=true, a3=true;
prime, a0=true, a1=true, a2=true, a3=true;
```

Transcript(Output):

```
(%i1)

t1: (not a3) and (not a2) and a1;
(%o1) ¬a3∧¬a2∧a1

(%i2)

t2: (not a3) and a2 and a0;
(%o2) ¬a3∧a2∧a0

(%i3)

t3: (not a2) and a1 and a0;
(%o3) ¬a2∧a1∧a0

(%i4)

t4: a2 and (not a1) and a0;
(%o4) a2∧¬a1∧a0

(%i5)

prime: t1 or t2 or t3 or t4;
(%o5) ¬a3∧¬a2∧a1∨¬a3∧a2∧a0∨¬a2∧a1∧a0∨a2∧¬a1∧a0

(%i6)

/* Testing all 16 combinations */
prime, a0=false, a1=false, a2=false, a3=false;
(%o6) false

(%i7)

prime, a0=true, a1=false, a2=false, a3=false;
(%o7) false

(%i8)

prime, a0=false, a1=true, a2=false, a3=false;
(%o8) true
```

```
(%i9)

prime, a0=true, a1=true, a2=false, a3=false;
(%o9) true

(%i10)

prime, a0=false, a1=false, a2=true, a3=false;
(%o10)      false

(%i11)

prime, a0=true, a1=false, a2=true, a3=false;
(%o11)      true

(%i12)

prime, a0=false, a1=true, a2=true, a3=false;
(%o12)      false

(%i13)

prime, a0=true, a1=true, a2=true, a3=false;
(%o13)      true

(%i14)

prime, a0=false, a1=false, a2=false, a3=true;
(%o14)      false

(%i15)

prime, a0=true, a1=false, a2=false, a3=true;
(%o15)      false

(%i16)

prime, a0=false, a1=true, a2=false, a3=true;
(%o16)      false
```

```
(%i17)

prime, a0=true, a1=true, a2=false, a3=true;
(%o17)      true

(%i18)

prime, a0=false, a1=false, a2=true, a3=true;
(%o18)      false

(%i19)

prime, a0=true, a1=false, a2=true, a3=true;
(%o19)      true

(%i20)

prime, a0=false, a1=true, a2=true, a3=true;
(%o20)      false

(%i21)

prime, a0=true, a1=true, a2=true, a3=true;
(%o21)      false
```

Results

Results for each combination can be obtained by running the above code in Maxima. Expected prime numbers among the 16 combinations (in decimal) are 2, 3, 5, and 7

Yellow Highlighted values from i8,i9,i11,i13,i17,i19 are prime numbers 2,3,5,7,11,13 respectively.
Green Highlighted values are the Prime Equation Output

Conclusion:

The truth table and Maxima results provides a view of the functionality of the prime number identifier circuit. It showcases the results for all 16 combinations of the 4-bit binary input, representing numbers from 0 to 15. We observed that the prime number identifier circuit correctly identifies the prime numbers within this range, which are 2, 3, 5, 7, 11, and 13. This demonstrates the effectiveness and accuracy of the circuit's logic in identifying prime numbers for the given range of inputs.