CP220 LAB 5, PRIME NUMBER IDENTIFIER CIRCUIT

AHMAD WALI

169036947

Wali6947@mylaurier.ca

Number	Binary(a1/a2/a3/a4)	**Prime**/Composite/None
0	0000	N
1	0001	N
2	0010	**P**
3	0011	**P**
4	0100	C
5	0101	**P**
6	0110	C
7	0111	**P**
8	1000	C
9	1001	C
10	1010	C
11	1011	**P**
12	1100	C
13	1101	**P**
14	1110	C
15	1111	C

We can determine if a number is prime or not, but realize that all numbers are either Composite, prime or neither (in rare cases). A composite number is a number that has 2 or more factors. A prime number is a number that has no factors other than 1 and itself. In this case, the numbers 0 and 1 must be neither of these, as they are special cases in the definitions above. We can check if a number is composite or prime by using an equation labeled below.

K MAP:

		$a_1 a_0$			
		00	01	11	10
$a_3 a_2$	00	0	0	1	1
	01	0	1	1	0
	11	0	1	0	0
	10	0	0	1	0

Looking for which values we get a prime number, we can finding prime numbers given the SOP equation:

$$prime = \overline{a_3}\ \overline{a_2}\ a_1 + \overline{a_3}\ a_2\ a_0 + \overline{a_2}\ a_1\ a_0 + a_2\ \overline{a_1}\ a_0$$

If we run this into the computer system Maxima, we get:

```
(%i10) prime, a0 = true, a1 = true, a2 = false, a3 = false;
```

(%o10) **true**

```
(%i11) prime, a0 = false, a1 = false, a2 = true, a3 = false;
```

(%o11) **false**

```
(%i12) prime, a0 = true, a1 = false, a2 = true, a3 = false;
```

(%o12) **true**

```
(%i13) prime, a0 = false, a1 = true, a2 = true, a3 = false;
```

(%o13) **false**

```
(%i14) prime, a0 = true, a1 = true, a2 = true, a3 = false;
```

(%o14) **true**

```
(%i15) prime, a0 = false, a1 = false, a2 = false, a3 = true;
```

(%o15) **false**

```
(%i16) prime, a0 = true, a1 = false, a2 = false, a3 = true;
```

(%o16) **false**

```
(%i17) prime, a0 = false, a1 = true, a2 = false, a3 = true;
```

(%o17) **false**

```
(%i18) prime, a0 = true, a1 = true, a2 = false, a3 = true;
```

(%o18) **true**

```
(%i1) prime = a3*a2*a1 + a3*a2*a0 + a2*a1*a0 + a2*a1*a0;
```

(%o1) $prime = a_1\, a_2\, a_3 + a_0\, a_2\, a_3 + 2\, a_0\, a_1\, a_2$

```
(%i2) t1: (not a3) and (not a2) and a1;
```

(%o2) $\neg a_3 \wedge \neg a_2 \wedge a_1$

```
(%i3) t2: (not a3) and a2 and a0;
```

(%o3) $\neg a_3 \wedge a_2 \wedge a_0$

```
(%i4) t3: (not a2) and a1 and a0;
```

(%o4) $\neg a_2 \wedge a_1 \wedge a_0$

```
(%i5) t4: a2 and (not a1) and a0;
```

(%o5) $a_2 \wedge \neg a_1 \wedge a_0$

```
(%i6) prime: t1 or t2 or t3 or t4;
```

(%o6) $\neg a_3 \wedge \neg a_2 \wedge a_1 \vee \neg a_3 \wedge a_2 \wedge a_0 \vee \neg a_2 \wedge a_1 \wedge a_0 \vee a_2 \wedge \neg a_1 \wedge a_0$

```
(%i7) prime, a0 = false, a1 = false, a2 = false, a3 = false;
```

(%o7) **false**

```
(%i8) prime, a0 = true, a1 = false, a2 = false, a3 = false;
```

(%o8) **false**

```
(%i9) prime, a0 = false, a1 = true, a2 = false, a3 = false;
```

(%o9) **true**

```
(%i19) prime, a0 = false, a1 = false, a2 = true, a3 = true;
```

(%o19) **false**

```
(%i20) prime, a0 = true, a1 = false, a2 = true, a3 = true;
```

(%o20) **true**

```
(%i21) prime, a0 = false, a1 = true, a2 = true, a3 = true;
```

(%o21) **false**

```
(%i22) prime, a0 = true, a1 = true, a2 = true, a3 = true;
```

(%o22) **false**

2,3,5,7,11,13 are all the prime numbers. These are the numbers that the system cannot determine if its composite (and knowing they cant be neither given the constraint of n>2), therefore implying that it must be prime.