Project Phase II REVISED: Digits of Pi — Conceptual Design Specification

Student: Shravya Rekhi
Student ID: 169087129
Course: CP-220
Date: October 31st 2025

Truth Table:

A3	A2	A1	A0	Decimal Place	e digit/digit	O3	O2	O1	O0
0	0	0	0	0	3	0	0	1	1
0	0	0	1	1	1	0	0	0	1
0	0	1	0	2	4	0	1	0	0
0	0	1	1	3	1	0	0	0	1
0	1	0	0	4	5	0	1	0	1
0	1	0	1	5	9	1	0	0	1
0	1	1	0	6	2	0	0	1	0
0	1	1	1	7	6	0	1	1	0
1	0	0	0	8	5	0	1	0	1
1	0	0	1	9	3	0	0	1	1
1	0	1	0	10	5	0	1	0	1
1	0	1	1	11	Invalid	0	0	0	0
1	1	0	0	12	Invalid	0	0	0	0
1	1	0	1	13	Invalid	0	0	0	0
1	1	1	0	14	Invalid	0	0	0	0
1	1	1	1	15	Invalid	0	0	0	0

Karnaugh Maps:

B_3	A_1A_0				
A_3A_2		00	01	11	10
	00	0	0	0	0
	01	0	1	0	0
	11	0	0	0	0
	10	0	0	0	0

B_2	A_1A_0				
A_3A_2		00	01	11	10
	00	0	0	0	1
	01	1	0	1	0
	11	0	0	0	0
	10	1	0	0	0

B_1	A_1A_0				
A_3A_2		00	01	11	10
	00	1	0	0	0
	01	0	0	1	1
	11	0	0	0	0
	10	0	1	0	0

B_0	A_1A_0				
A_3A_2		00	01	11	10
	00	1	1	1	0
	01	1	1	0	0
	11	0	0	0	0
	10	1	1	0	0

Finalized simplified equations (from truth table minimization):

$O3 = \neg A3 \cdot A2 \cdot \neg A1 \cdot \neg A0$

$O2 = (\neg A3 \cdot A1) + (A3 \cdot \neg A2 \cdot A0)$

$O1 = (\neg A3 \cdot \neg A2 \cdot \neg A1 \cdot A0) + (\neg A3 \cdot A2 \cdot A1) + (A3 \cdot \neg A2 \cdot \neg A0)$

$O0 = (\neg A3 \cdot \neg A2 \cdot A1 \cdot A0) + (\neg A3 \cdot A2 \cdot \neg A1 \cdot \neg A0) + (A3 \cdot \neg A2 \cdot \neg A0)$

Testing Equation:

Input:

```
/* Boolean equations for each output bit of the π digit
display */

o3: not(a3) and a2 and not(a1) and not(a0);
o2: (not(a3) and a1) or (a3 and not(a2) and a0);
o1: (not(a3) and not(a2) and not(a1) and a0)
     or (not(a3) and a2 and a1)
     or (a3 and not(a2) and not(a0));
o0: (not(a3) and not(a2) and a1 and a0)
     or (not(a3) and a2 and not(a1) and not(a0))
     or (a3 and not(a2) and not(a0));

/* Final π digit output in binary = o3o2o1o0 */

/* Test all valid inputs (0–10) */
[o3, o2, o1, o0],
a3=false,a2=false,a1=false,a0=false;  /* 0 */
[o3, o2, o1, o0],
a3=false,a2=false,a1=false,a0=true;   /* 1 */
[o3, o2, o1, o0],
a3=false,a2=false,a1=true,a0=false;   /* 2 */
[o3, o2, o1, o0],
a3=false,a2=false,a1=true,a0=true;    /* 3 */
[o3, o2, o1, o0],
a3=false,a2=true,a1=false,a0=false;   /* 4 */
[o3, o2, o1, o0],
a3=false,a2=true,a1=false,a0=true;    /* 5 */
[o3, o2, o1, o0],
a3=false,a2=true,a1=true,a0=false;    /* 6 */
[o3, o2, o1, o0],
a3=false,a2=true,a1=true,a0=true;     /* 7 */
[o3, o2=false,a1=false,a0=false];     /* 8 */
[o3, o2=false,a1=false,a0=true];      /* 9 */
[o3, o2=true,a1=false,a0=false];      /* 10 */
```

Output:

$$\neg a_3 \wedge a_2 \wedge \neg a_1 \wedge \neg a_0$$

$$\neg a_3 \wedge a_1 \vee a_3 \wedge \neg a_2 \wedge a_0$$

$$\neg a_3 \wedge \neg a_2 \wedge \neg a_1 \wedge a_0 \vee \neg a_3 \wedge a_2 \wedge a_1 \vee a_3 \wedge \neg a_2 \wedge \neg a_0$$

$$\neg a_3 \wedge \neg a_2 \wedge a_1 \wedge a_0 \vee \neg a_3 \wedge a_2 \wedge \neg a_1 \wedge \neg a_0 \vee a_3 \wedge \neg a_2 \wedge \neg a_0$$

[**false**, **false**, **false**, **false**]

[**false**, **false**, **true**, **false**]

[**false**, **true**, **false**, **false**]

[**false**, **true**, **false**, **true**]

[**true**, **false**, **false**, **true**]

[**false**, **false**, **false**, **false**]

[**false**, **true**, **true**, **false**]

[**false**, **true**, **true**, **false**]

$[\neg a_3 \wedge a_2 \wedge \neg a_1 \wedge \neg a_0, (\neg a_3 \wedge a_1 \vee a_3 \wedge \neg a_2 \wedge a_0) = \mathbf{false}, a_1 = \mathbf{false}, a_0 = \mathbf{false}]$

$[\neg a_3 \wedge a_2 \wedge \neg a_1 \wedge \neg a_0, (\neg a_3 \wedge a_1 \vee a_3 \wedge \neg a_2 \wedge a_0) = \mathbf{false}, a_1 = \mathbf{false}, a_0 = \mathbf{true}]$

$[\neg a_3 \wedge a_2 \wedge \neg a_1 \wedge \neg a_0, (\neg a_3 \wedge a_1 \vee a_3 \wedge \neg a_2 \wedge a_0) = \mathbf{true}, a_1 = \mathbf{false}, a_0 = \mathbf{false}]$

incorrect syntax: Premature termination of input at ;. /* 10 */; ^

Summary:

A 4-bit binary input (representing decimal places 0–10) is mapped to a 4-bit binary output (representing the appropriate digit of π) via the Pi Digit Circuit, a combinational logic circuit. When an input falls outside of the acceptable range (decimal 11–15), it is regarded as a don't-care situation and produces an output of 0000 to denote an error or invalid input.

The circuit's outputs were reduced to effective logic equations using Karnaugh maps, which minimized the number of gates needed while guaranteeing accurate representation of π digits. These simplified equations are used to generate each output bit (O3–O0), ensuring accurate results for all valid inputs.

This design produces a dependable, verifiable, and easily implementable representation of the first eleven digits of π by demonstrating a practical use of combinational logic, K-map simplification, and digital encoding.