Design Document

Tart888

ตู้สล็อตแมชชีนอัจฉริยะ

จัดทำโดย

1. นาย ราธา โรจน์รุจิพงศ์ รหัสนักศึกษา 66011464 กลุ่ม 17
2. นาย วัฒน์นันท์ ธีรธนาพงษ์ รหัสนักศึกษา 66011476 กลุ่ม 17

นำเสนอ

รศ.ดร. เจริญ วงษ์ชุ่มเย็น

โครงงานนี้เป็นส่วนหนึ่งของรายวิชา 01076112 DIGITAL SYSTEM FUNDAMENTALS

ภาควิชาวิศวกรรมศาสตร์สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 1 ปีการศึกษา 2567

สารบัญ

บทนำ	1
โครงสร้างโครงงาน	2
อุปกรณ์ที่ใช้	3
หลักการทำงาน	6
State Diagram	9
แผนการดำเนินงาน	11
Flowchart	12
Top-Down Design	13
Invention	21

ที่มาและความสำคัญ

ในปัจจุบัน FPGA เป็นเทคโนโลยีที่สำคัญในด้านการพัฒนาระบบดิจิทัล การจำลองตู้สล็อตแมชชีนด้วย FPGA จึงเป็นโครงการที่น่าสนใจ เพราะช่วยให้เข้าใจการทำงานภายในของเครื่องเกมที่มีการใช้ระบบการสุ่มและการแสดงผล

วัตถุประสงค์ในการทำโครงงาน

1. เพื่อประยุกต์ใช้ความรู้จากวิชา Digital system fundamental
2. เพื่อศึกษาการทำงานของบอร์ด FPGA และ โปรแกรม Xilinx และการสร้าง Schematic Module และ VHDL Module
3. เพื่อพัฒนาเครื่องเล่น Slot machine
4. เพื่อเป็นแนวทางในการจัดทำโครงงานแก่ผู้อื่นสืบต่อไป

ประโยชน์ที่ได้รับจากการทำโครงงาน

1. ได้ความชำนาญและประสบการณ์ในศาสตร์ด้าน Digital
2. ได้ความชำนาญและประสบการณ์ในการใช้งานบอร์ด FPGA
3. ได้ความชำนาญและประสบการณ์ในการใช้งานโปรแกรม Xilinx และการสร้าง Schematic Module และ VHDL Module
4. ได้ความชำนาญและประสบการณ์ในการสร้าง Top-Down Design
5. สามารถนำชิ้นงานไปต่อยอดเพิ่มในอนาคตได้

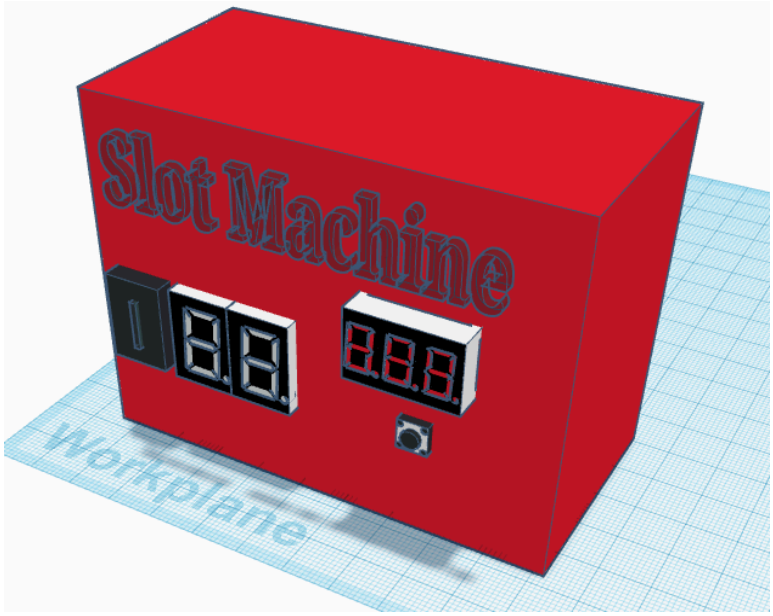
ขอบเขตของโครงงาน

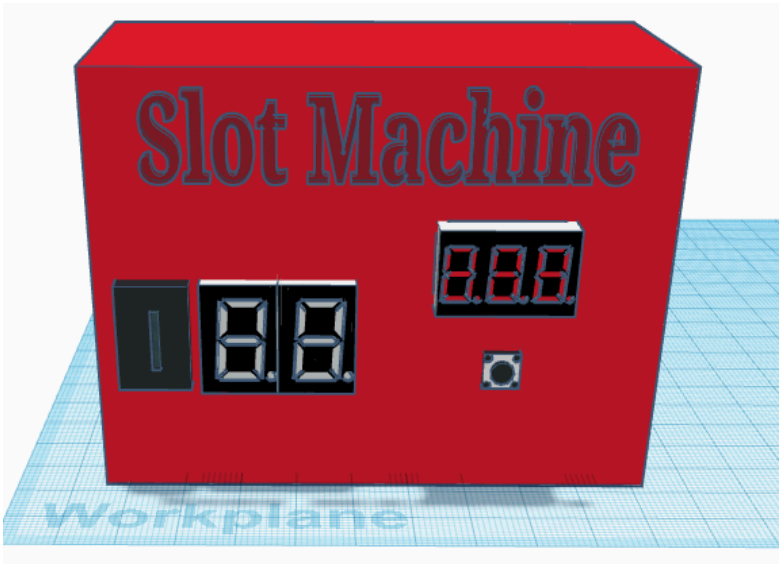
1. การสร้างวงจรใน FPGA Spartan-6 ของ Xilinx เบอร์ XC6SLX9
2. การเขียนโปรแกรมของบอร์ด Arduino IDE 2.3.2

ระยะเวลาในการพัฒนาโครงงาน

ระยะเวลาในการพัฒนาโครงงานทั้งหมด 43 วัน โดยเริ่มตั้งแต่วันที่ 1 ตุลาคม พ.ศ. 2567 ถึง 12 พฤศจิกายน พ.ศ. 2567

โครงสร้างของโครงงาน

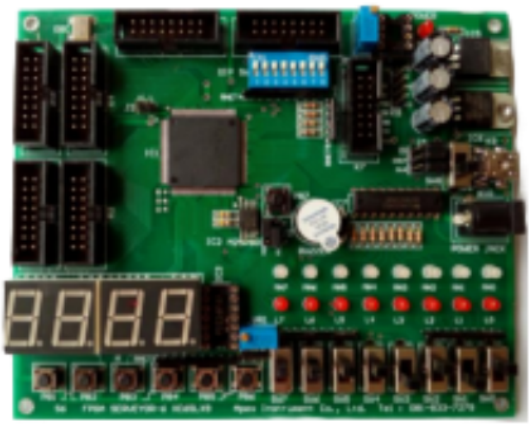




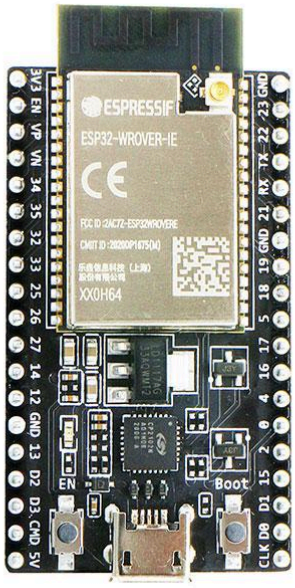
รูปแสดงโมเดลตู้สล็อตแมชชีน

อุปกรณ์ที่ใช้

1. รุ่น FPGA Spartan-6 ของ Xilinx เบอร์ XC6SLX9 จำนวน 2 ตัว



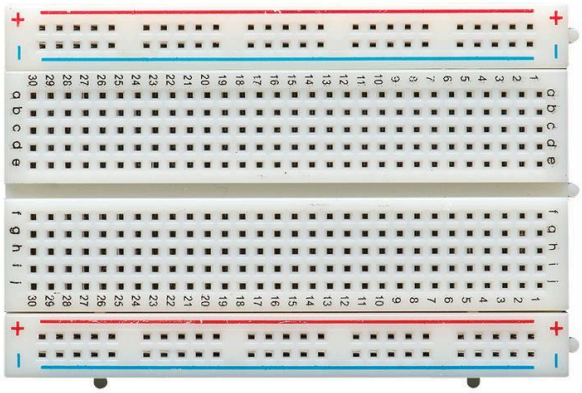
2. ESP32 WiFi



3. Coin Validator

4. สายไฟเชื่อม

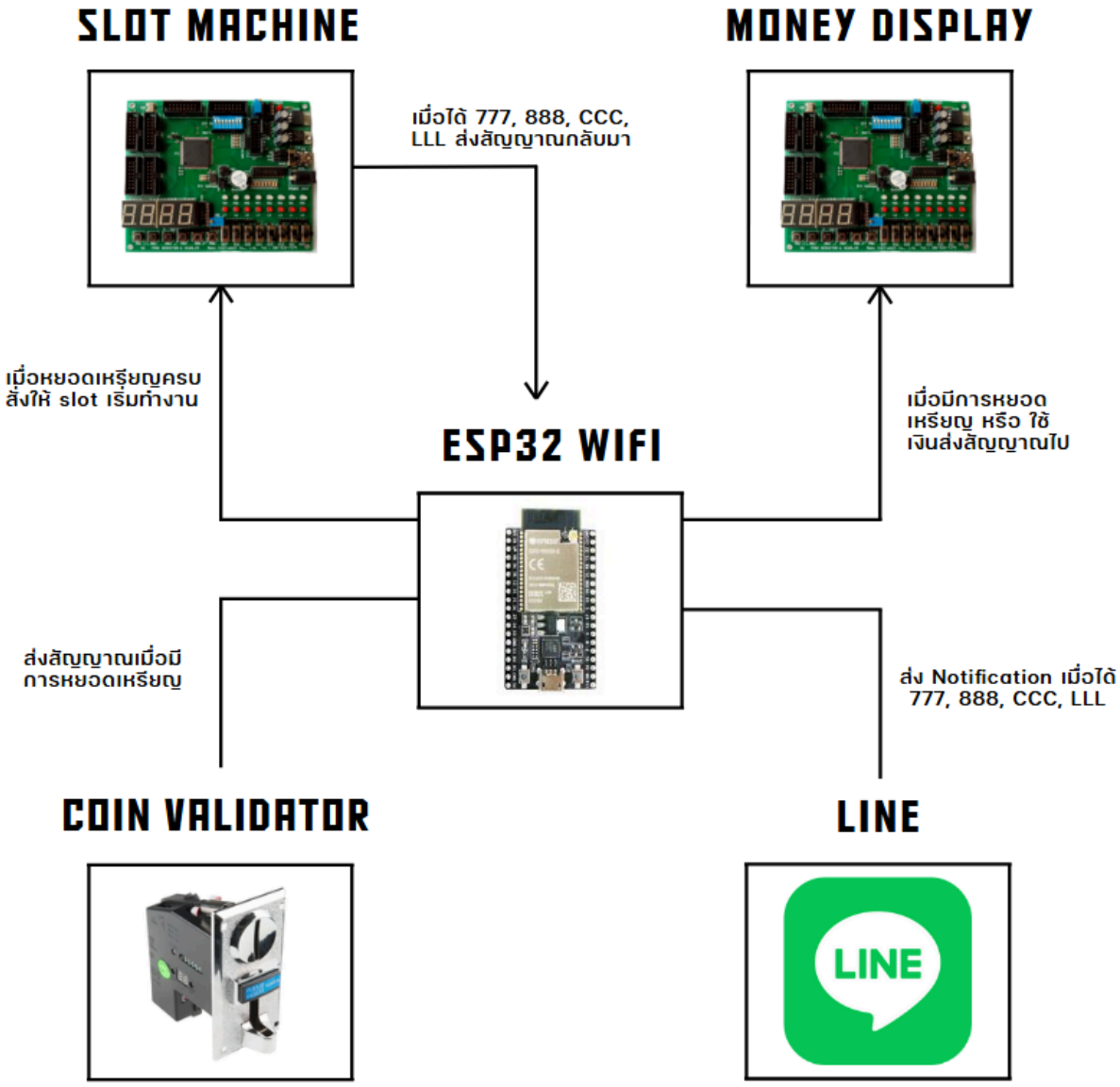
5. Bread Board



6. Push Button

7. Battery 12V

หลักการทำงาน



รูปแสดงหลักการทำงานโดยรวมของโปรแกรม

1. FPGA บอร์ด 1 (เครื่อง Slot Machine)

- **หน้าที่**: ควบคุมสถานะ (State) ในการเล่นและหมุนช่องแสดงผล 3 ช่องของ Slot Machine
- **การทำงาน**:
    - ระบบ Slot Machine จะรันบนบอร์ดนี้ โดยแสดงผลของทั้ง 3 ช่องบน 7 Segment Display
    - 7 Segment Decoder เป็นแบบ Custom ที่ออกแบบมาให้แสดงสัญลักษณ์พิเศษ เช่น 7, 8, C, L และ X
    - หากผู้เล่นชนะตามรูปแบบคะแนน เช่น 777, 888, CCC, LLL: จะส่งสัญญาณ 1 ไปยังบอร์ด ESP32 เพื่อส่งแจ้งเตือนเข้าไลน์
    - FPGA จะรอรับสัญญาณเริ่มต้นการทำงานจาก Arduino เมื่อมีการหยอดเหรียญครบตามที่กำหนด

2. FPGA บอร์ด 2 (แสดงผลจำนวณเงิน)

- **หน้าที่**: รอรับสัญญาณจาก ESP32 เมื่อมีการหยอดเหรียญเพื่อทำการเพิ่มจำนวนเงิน
- **การทำงาน**:
    - มี 7 สายสัญญาณสำหรับรับเลขฐาน 2 (7-Bit) จาก ESP32
    - นำเลขฐาน 2 (7-Bit) มาแสดงผลเป็นฐาน 10 บน 7 Segment Display

3. ESP 32 Wifi

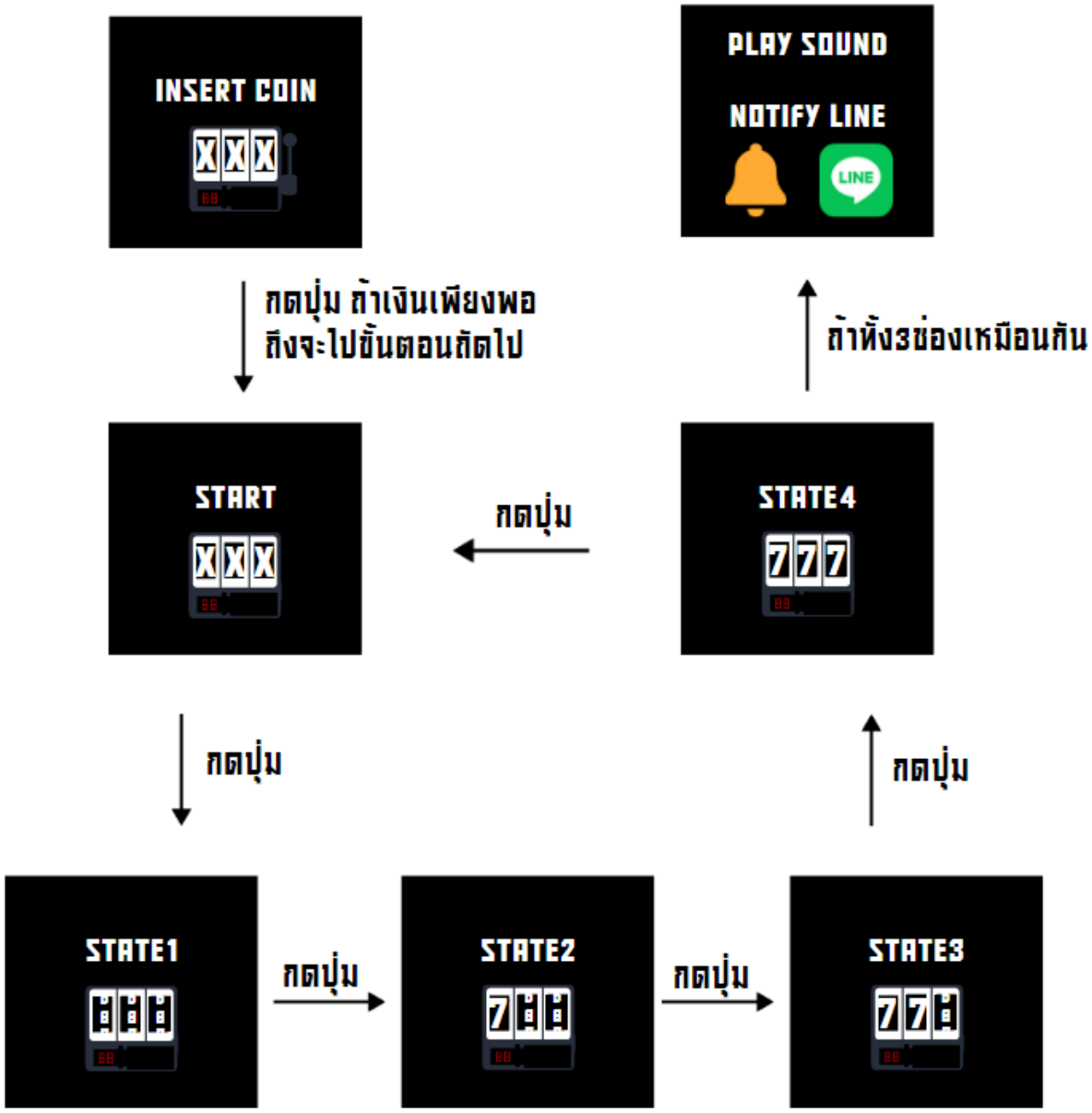
- **หน้าที่**: รอรับสัญญาณจากบอร์ดที่ 1 และ Coin Validator เพื่อแจ้งเตือนไปยัง LINE
- **การทำงาน**:
    - รอรับสัญญาณจาก Coin Validator เพื่อเพิ่มจำนวนเงิน
    - ส่งจำนวนเงินไปยังบอร์ด2 โดยส่งเป็นเลขฐาน 2 (7-Bit)
    - รอรับสัญญาณคะแนนจาก FPGA บอร์ดที่ 1 โดยมี 4 สายสัญญาณตามรูปแบบคะแนนที่ชนะ:
        - สายที่ 1: คะแนนแบบ 777
        - สายที่ 2: คะแนนแบบ 888
        - สายที่ 3: คะแนนแบบ CCC
        - สายที่ 4: คะแนนแบบ LLL
    - แจ้งเตือนผ่าน Line ตามคะแนนที่ได้

4. Coin Validator

- **หน้าที่**: ตรวจสอบเหรียญที่หยอดเข้ามาในตู้ Slot Machine
- **การทำงาน**:

- เมื่อมีการหยอดเหรียญ Coin Validator จะส่งสัญญาณไปยัง ESP32 เพื่อเพิ่มจำนวนเงิน

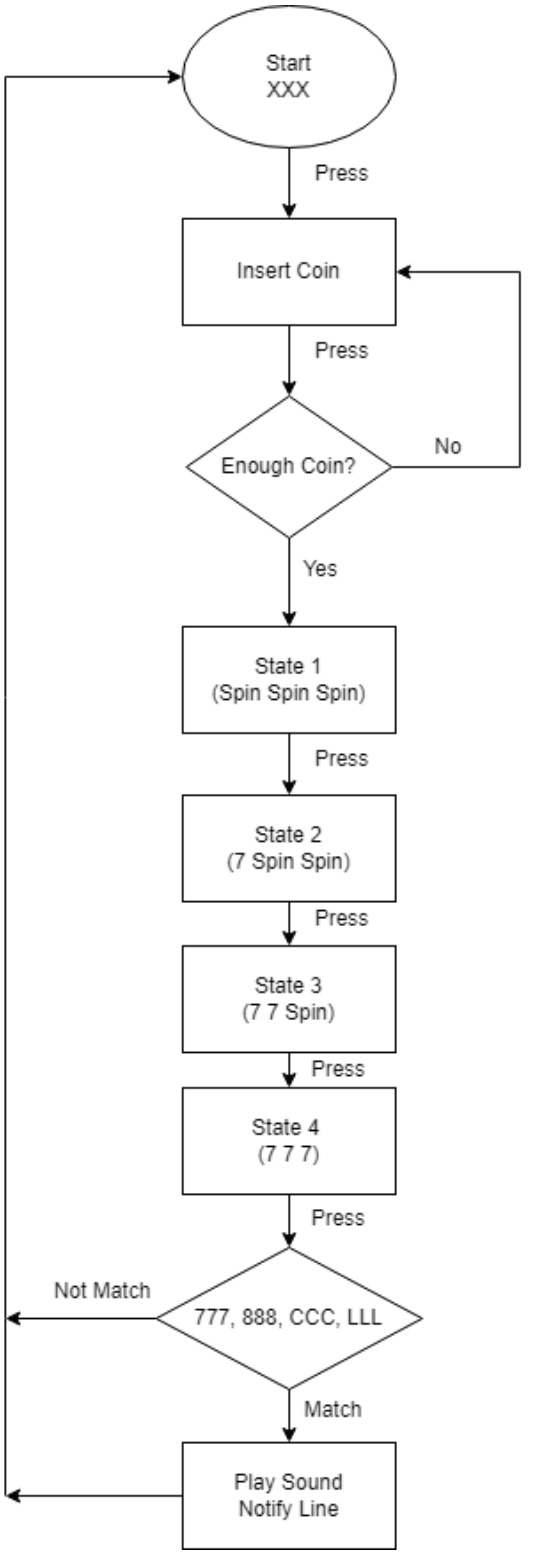
State Diagram



แผนการดำเนินงาน

รายละเอียดขั้นตอนทำโครงงาน	ระยะเวลาการดำเนินงาน						
	1-7 ต.ค.	8-14 ต.ค.	15-21 ต.ค.	22-28 ต.ค.	29 ต.ค. - 4 พ.ย.	5 - 11 พ.ย.	12 - 18 พ.ย.
คันคว้าและศึกษาหาหัวนั้นที่ สนใจในการทำโครงงาน	7						
ศึกษาหลักการทำเกมบอร์ด FPGA	14						
ออกแบบลักษณะเกมว่าอยากให้มีอะไรบ้าง		14					
ทำการสั่งซื้ออุปกรณ์ที่เกี่ยวของกับโครงงาน			7				
ทำการเขียนวงจรและทดสอบ				21			
จัดทำ VDO นำเสนอ							7

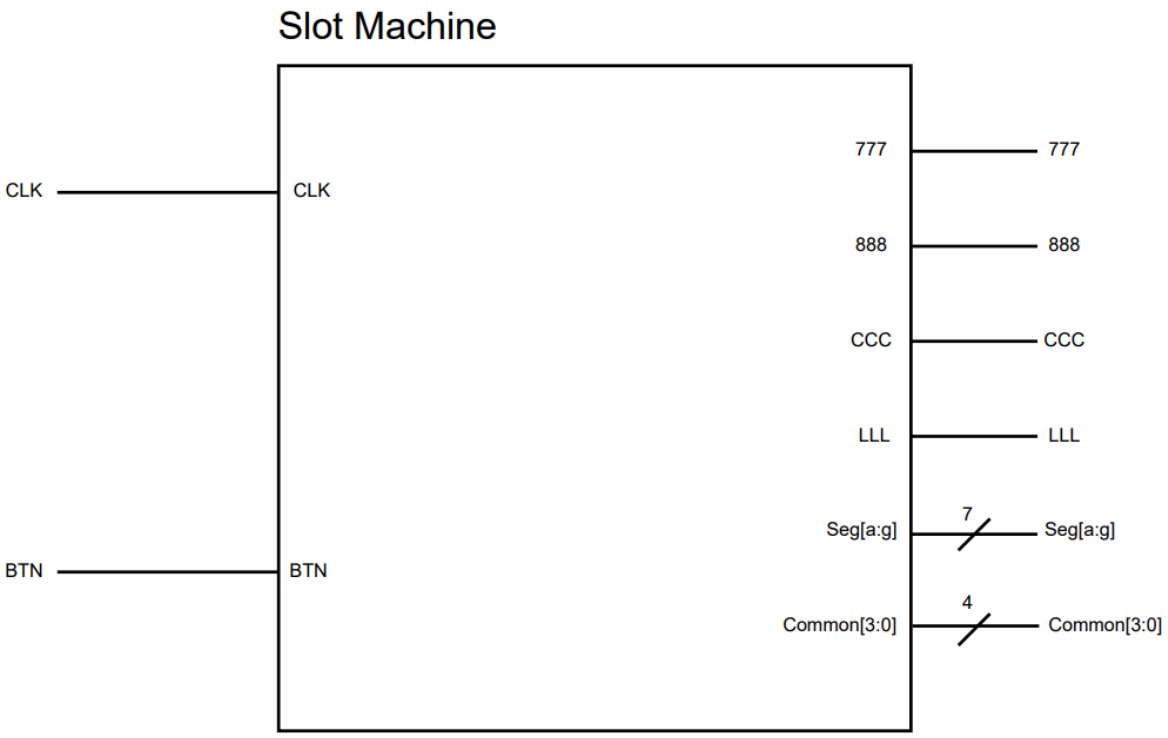
Flowchart



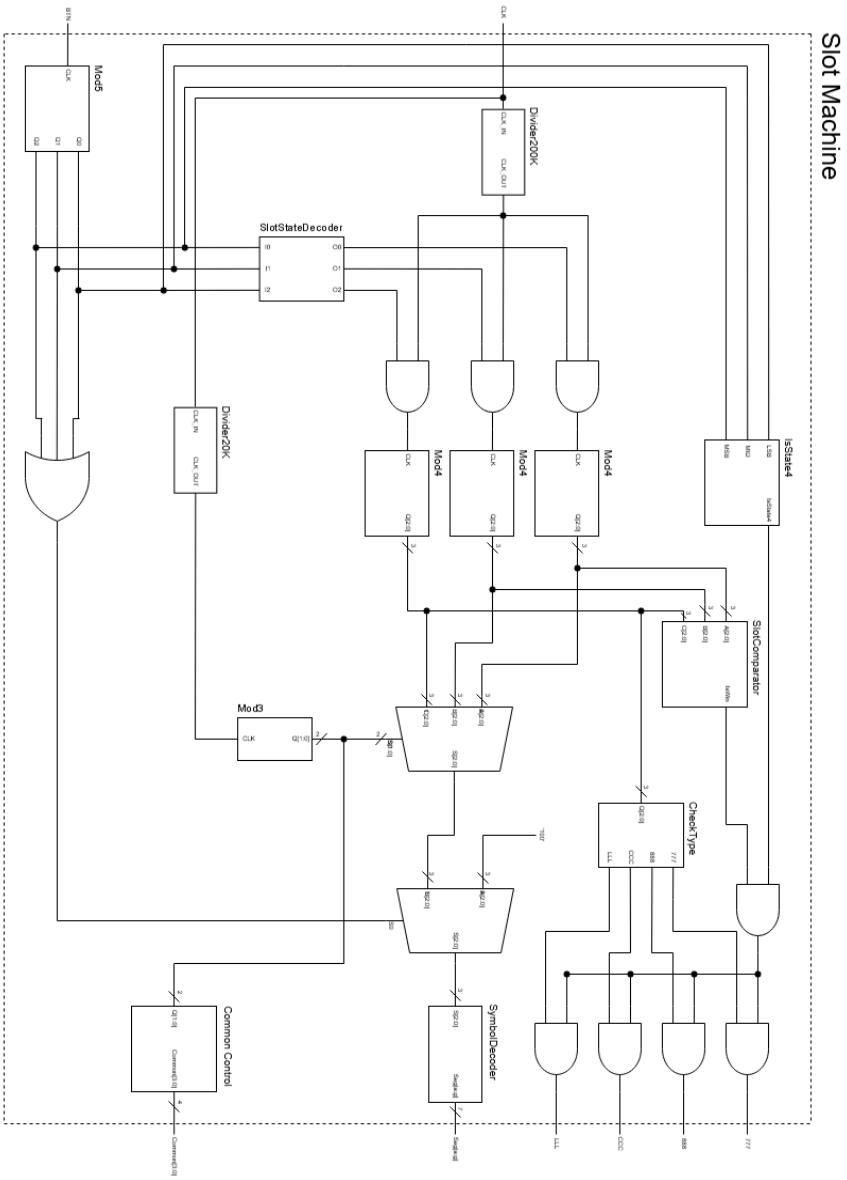
รูปแสดง Flowchart การทำงานโครงงานนี้

Top-Down Design (Slot Machine)

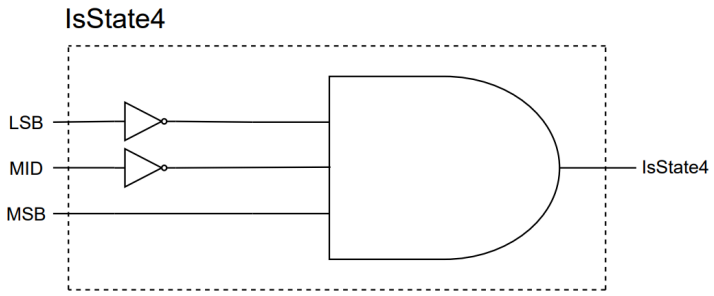
Top Layer



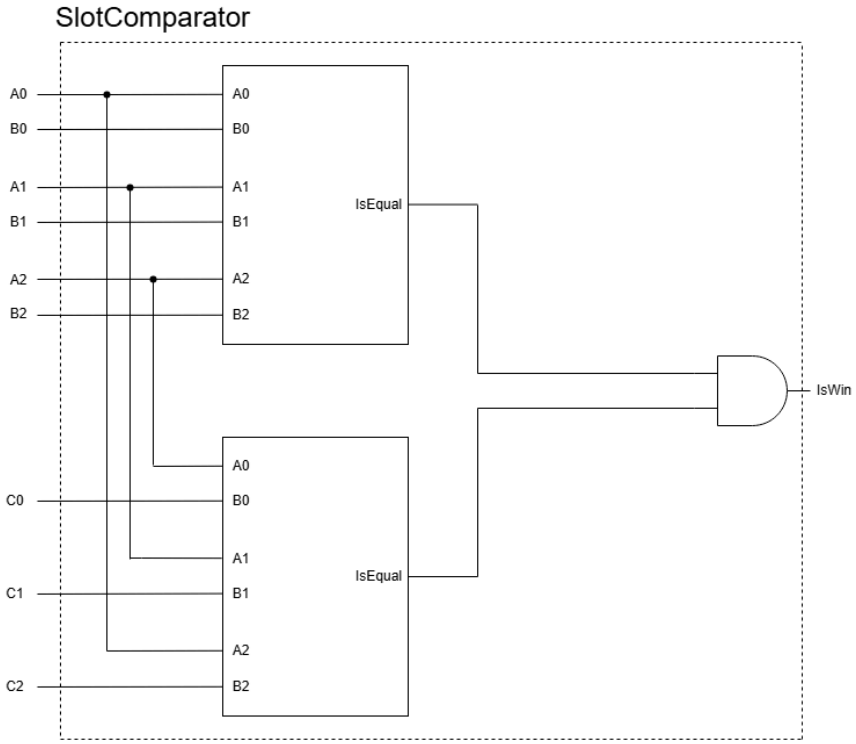
2nd Layer (Slot Machine)



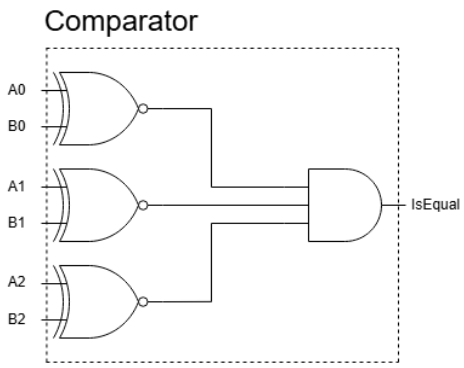
3rd Layer (IsState4)



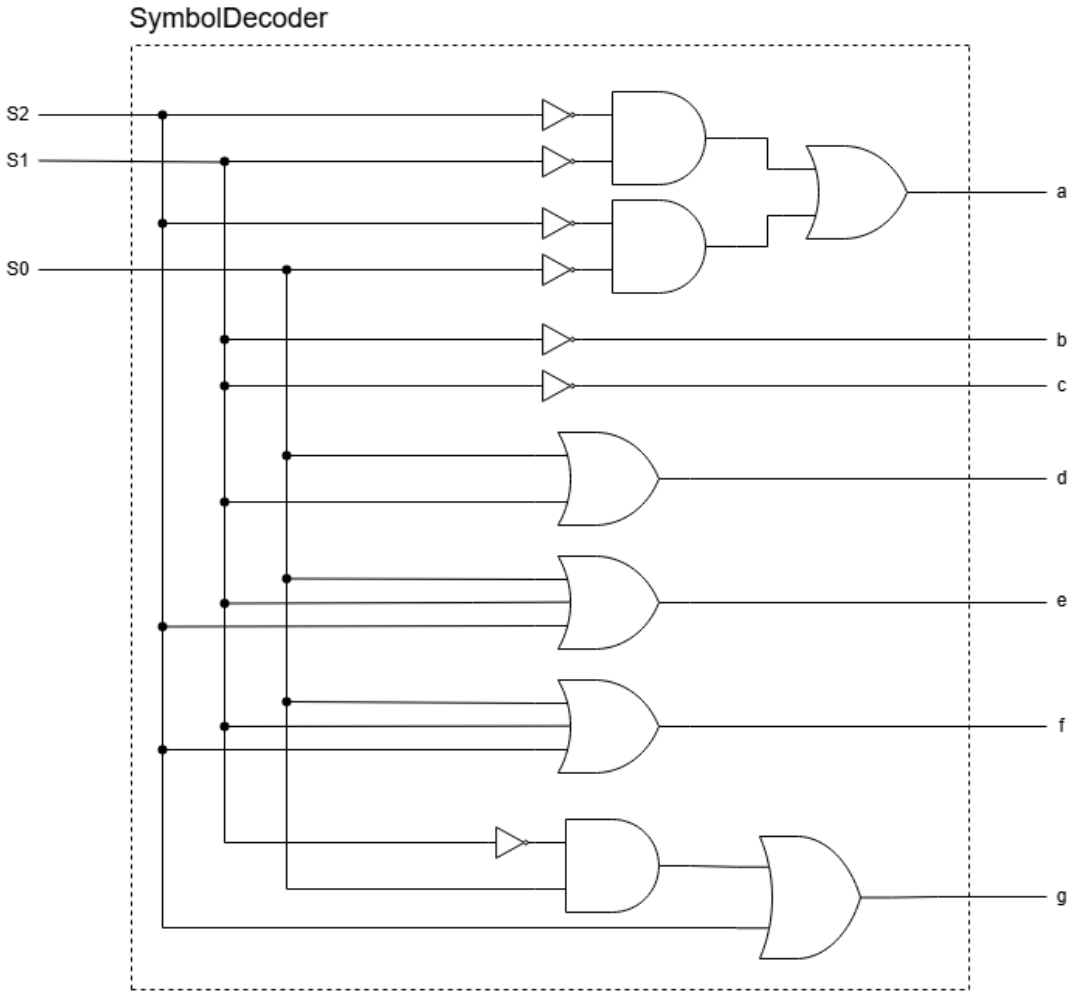
3rd Layer (SlotComparator)



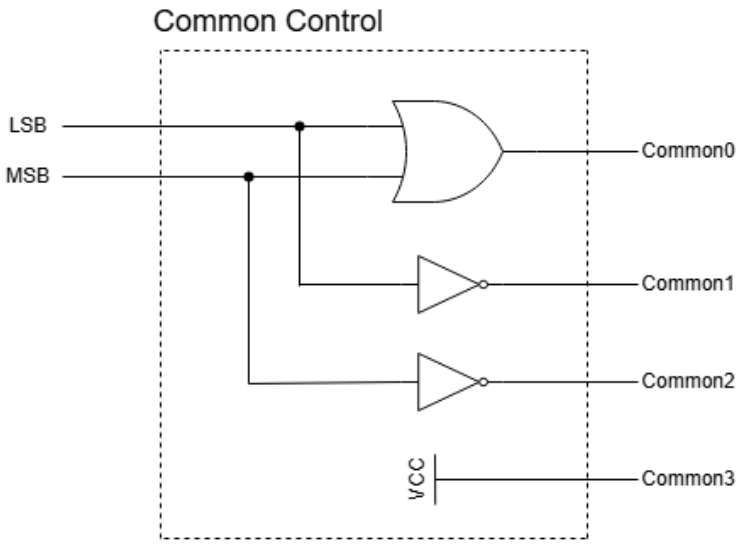
4th Layer (Comparator)



3rd Layer (SymbolDecoder)



3rd Layer (CommonControl)



3rd Layer (SlotDecoder)

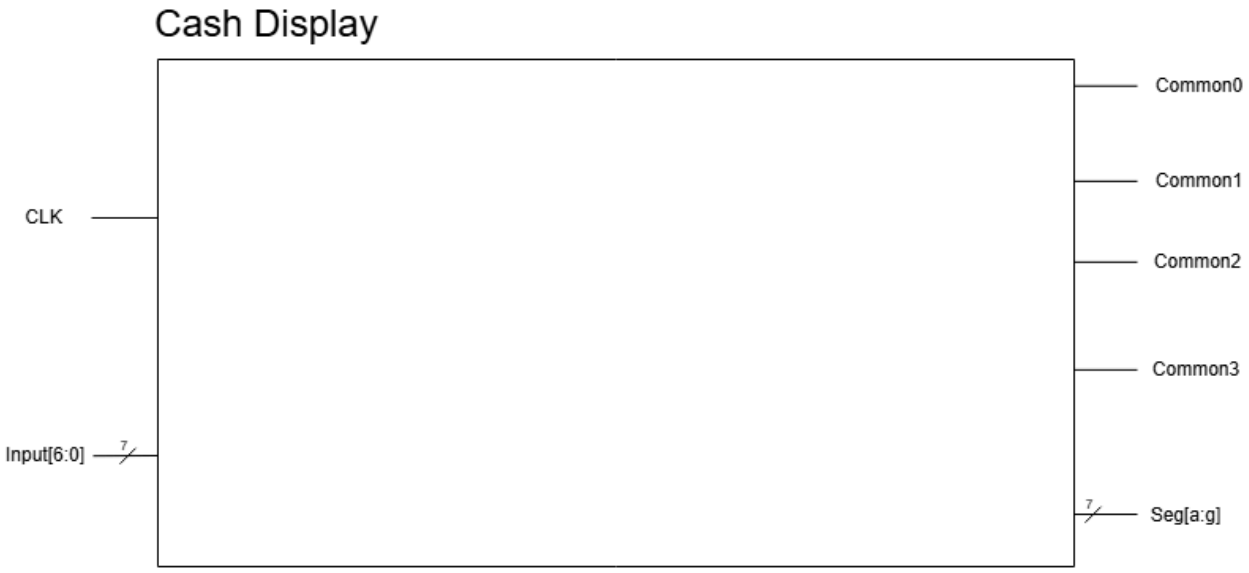
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity SlotDecoder is
    Port (
        I2  : in  STD_LOGIC;  -- 3-bit input
        I1  : in  STD_LOGIC;
        I0  : in  STD_LOGIC;
        O0  : out STD_LOGIC;  -- Output 0
        O1  : out STD_LOGIC;  -- Output 1
        O2  : out STD_LOGIC   -- Output 2
    );
end SlotDecoder;

architecture Behavioral of SlotDecoder is
    signal input_bits : STD_LOGIC_VECTOR(2 downto 0);  -- 3-bit vector to hold input
begin
    input_bits <= I2 & I1 & I0;  -- Concatenate inputs into a 3-bit vector

    process (input_bits)
    begin
        case input_bits is
            when "000" =>
                O0 <= '0';
                O1 <= '0';
                O2 <= '0';
            when "001" =>
                O0 <= '1';
                O1 <= '1';
                O2 <= '1';
            when "010" =>
                O0 <= '0';
                O1 <= '1';
                O2 <= '1';
            when "011" =>
                O0 <= '0';
                O1 <= '0';
                O2 <= '1';
            when "100" =>
                O0 <= '0';
                O1 <= '0';
                O2 <= '0';
            when others =>
                O0 <= '0';
                O1 <= '0';
                O2 <= '0';  -- Default case
        end case;
    end process;

end Behavioral;
```

3rd Layer (CheckType)

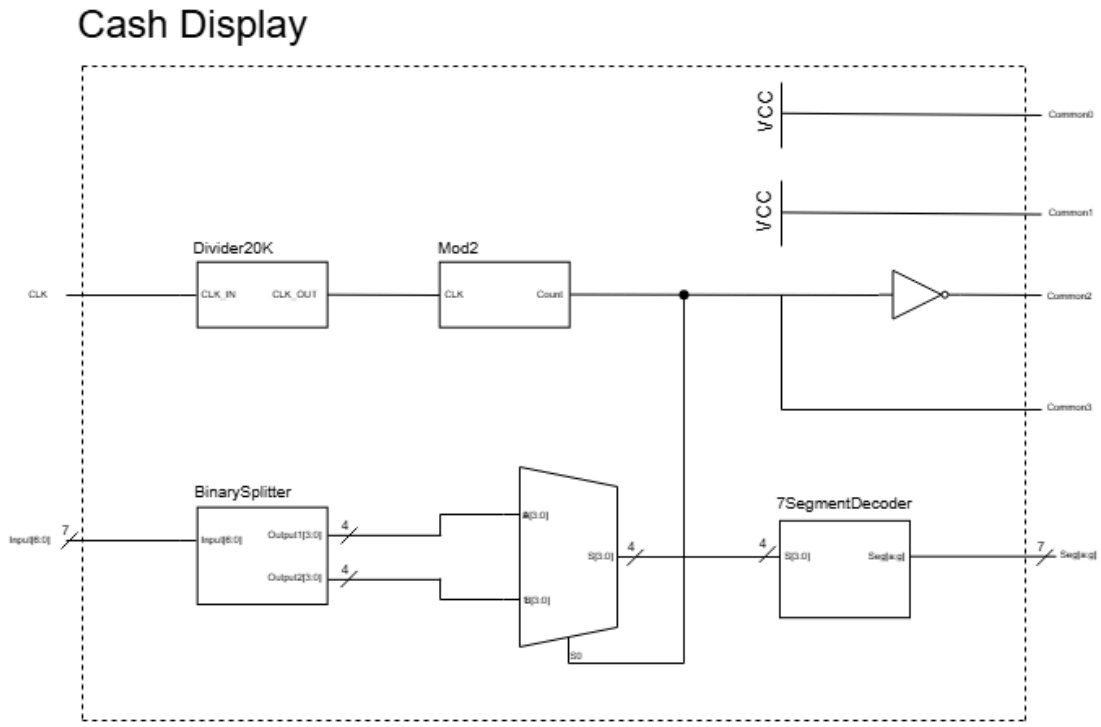
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity CheckType is
    Port (
        q0      : in  STD_LOGIC;   -- Input bit 0
        q1      : in  STD_LOGIC;   -- Input bit 1
        q2      : in  STD_LOGIC;   -- Input bit 2
        is0     : out STD_LOGIC;   -- Output for 000
        is1     : out STD_LOGIC;   -- Output for 001
        is2     : out STD_LOGIC;   -- Output for 010
        is3     : out STD_LOGIC    -- Output for 011
    );
end CheckType;

architecture Behavioral of CheckType is
    signal input_bits : STD_LOGIC_VECTOR(2 downto 0);
begin
    -- Concatenate inputs to form a 3-bit vector
    input_bits <= q2 & q1 & q0;

    process(input_bits)
    begin
        -- Set all outputs to 0 by default
        is0 <= '0';
        is1 <= '0';
        is2 <= '0';
        is3 <= '0';

        -- Check input combinations and set corresponding output
        case input_bits is
            when "000" =>
                is0 <= '1';
            when "001" =>
                is1 <= '1';
            when "010" =>
                is2 <= '1';
            when "011" =>
                is3 <= '1';
            when others =>
                -- No output is set to 1 for other cases
                null;
        end case;
    end process;
end Behavioral;
```

Top-Down Design (Cash Display)

Top Layer



2nd Layer (Cash Display)



3rd Layer (Binary Splitter)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity BinarySplitter is
    Port (
        input_bin  : in  STD_LOGIC_VECTOR(6 downto 0);  -- 7-bit binary input
        output1    : out STD_LOGIC_VECTOR(3 downto 0);  -- 4-bit binary output for first decimal digit
        output2    : out STD_LOGIC_VECTOR(3 downto 0)   -- 4-bit binary output for second decimal digit
    );
end BinarySplitter;

architecture Behavioral of BinarySplitter is
    signal decimal1, decimal2 : INTEGER range 0 to 15;  -- Signals to hold decimal values
begin
    process(input_bin)
        variable bin_value : INTEGER;
    begin
        -- Convert input binary to an integer
        bin_value := CONV_INTEGER(input_bin);

        -- Calculate the decimal values
        decimal1 <= bin_value / 10;  -- Get the first decimal digit (most significant)
        decimal2 <= bin_value mod 10;  -- Get the second decimal digit (least significant)

        -- Convert decimal values to 4-bit binary outputs
        output1 <= CONV_STD_LOGIC_VECTOR(decimal1, 4);
        output2 <= CONV_STD_LOGIC_VECTOR(decimal2, 4);
    end process;
end Behavioral;
```

Invention


10฿
PLEASE INSERT COIN
COIN REJECT BAR
Amount:
10 ฿
TART
777
LLL 10 p.
CCC 20 p.
888 50 p.
777 100 p.
Play here!