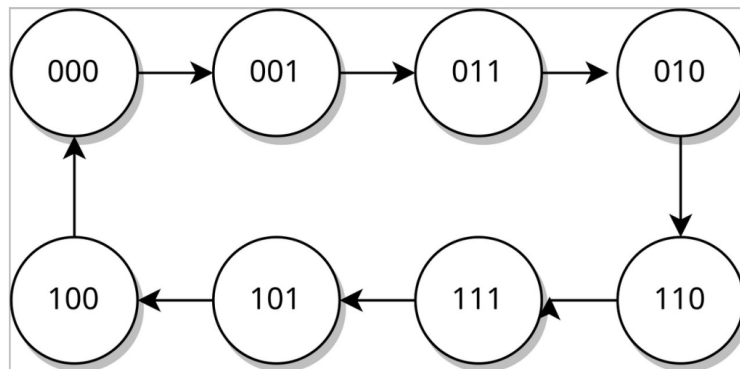


## Problem 1: Counter

จงสร้าง Counter ที่มีสัญญาณเข้าเป็น clock (**clk**) 1 ตัว, และ สัญญาณ input **reset** และมี output เป็น A, B และ C ซึ่งมี output ดังต่อไปนี้



โดยกำหนดให้ เมื่อ reset = 1 ค่าจะเริ่มต้นที่ CBA = 000, (reset เป็น synchronous) เมื่อนำค่า reset = 0, CBA จะเปลี่ยนเป็น 001, 011, 010, 110, 111, 101, 100 แล้ววนมาที่ 000 ตามลำดับ

ตัวอย่าง Testcase อยู่ใน template\_01.dig

### ข้อมูลนำเข้า

- Input: reset
- Clock: clk

### ข้อมูลส่งออก

- Output: A, B, และ C

## Problem 2: Square Root

สร้างวงจรหารากที่สองของ input โดยให้ input เป็นเลข **A** 16 บิต, **start** 1 บิต และมี output **sqrt** 8 บิตเป็นค่ารากที่สองของ **A** ที่เป็นจำนวนเต็มมากที่สุดที่น้อยกว่าค่ารากที่สองของ **A** (ตัวอย่างเช่น **A** = 10, **sqrt**= 3)

วงจรจะเริ่มทำงานเมื่อมีการเปลี่ยน **start** จาก 0 เป็น 1 ระหว่างการทำงานให้ **busy** เป็น 1 และเมื่อคำนวณเสร็จสิ้นแล้วให้แสดงค่ารากที่สองใน **sqrt** และให้ค่า **busy** กลับเป็น 0

**หมายเหตุ 1** : ช่วงระหว่างคำนวณอยู่ไม่ต้องสนใจสัญญาณ **start** และให้แสดงค่ารากที่สองของ **A** ค้างไว้จนกว่าจะมีการเปลี่ยน **start** จาก 0 เป็น 1 ใหม่จึงสามารถเปลี่ยนค่าได้

**หมายเหตุ 2** : input **A** มีค่าตั้งแต่ 0 เป็นต้นไป และจะไม่เปลี่ยนแปลงในช่วงที่ **busy** เป็น 1

**หมายเหตุ 3** : ตัวตรวจจะรอสัญญาณไม่เกิน 500 cycles (ดังตัวอย่างใน **template\_02.dig**)

**Hint**: สามารถใช้ อุปกรณ์ **Multiply** หรือ **Division** ซึ่งอยู่ใน Arithmetic ได้

ตัวอย่าง Testcase อยู่ใน **template\_02.dig**

### ข้อมูลนำเข้า

- Input: **A** ขนาด 16 Bit
- Input: **start** ขนาด 1 Bit
- Clock: **clk**

### ข้อมูลส่งออก

- Output: **sqrt** ขนาด 8 Bit
- Output: **busy** ขนาด 1 Bit

### Problem 3: Mean

จงสร้างวงจรหาค่าเฉลี่ยของตัวเลข โดยสมมุติสามารถรับค่าได้ผ่านทาง input เป็นเลขฐานสอง **A** 16 บิต, **add** 1 บิตและ **reset** 1 บิต และมี output เป็นเลขฐานสอง **mean** 16 บิตและ **busy** 1 บิต

เมื่อ **add** มีค่าเปลี่ยนจาก 0 เป็น 1 ให้เก็บค่าจาก **A** และแสดงค่าเฉลี่ยของตัวเลขใน **mean** (ปัดเศษลง) หากวงจรใช้เวลาในการหาค่าเฉลี่ยให้มีค่า **busy** เป็น 1 จนกระทั่งทำงานเสร็จ ถ้า **reset** เป็น 1 จะทำการเริ่มต้นวงจรใหม่และวงจรจะทำงานก็ต่อเมื่อ **reset** เป็น 0

หลังจาก **reset** ในแต่ละครั้งจะมีการ **add** ไม่เกิน 15 ครั้ง

**หมายเหตุ 1 :** input **A** จะไม่เปลี่ยนแปลงในช่วงที่ **busy** เป็น 1

**หมายเหตุ 2 :** จะรับประกันว่า **reset** จะเป็น 1 อย่างน้อย 1 cycle เมื่อเริ่มโปรแกรม

**หมายเหตุ 3 :** ตัวตรวจจะรอสัญญาณไม่เกิน 1000 cycles ต่อการรับค่าเพิ่มหนึ่งค่า (ดังตัวอย่างใน **template\_03.dig**)

**Hint:** สามารถใช้ อุปกรณ์ **Multiply** หรือ **Division** ซึ่งอยู่ใน Arithmetic ได้

ตัวอย่าง Testcase อยู่ใน **template\_03.dig**

#### ข้อมูลนำเข้า

- A ขนาด 16 Bit
- input ขนาด 1 Bit
- add ขนาด 1 Bit

#### ข้อมูลส่งออก

- mean ขนาด 16 Bit
- busy ขนาด 1 Bit